Amazon EMR

Management Guide

Amazon EMR: Management Guide

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Qu'est-ce qu'Amazon EMR ? 1
- Présentation 1
  - Présentation des clusters et des nœuds 1
  - Soumission de travail à un cluster 2
  - Traitement des données 2
  - Présentation du cycle de vie du cluster 4
- Avantages 4
  - Economies sur les coûts 5
  - Intégration à AWS 5
  - Déploiement 5
  - Evolutivité et flexibilité 6
  - Fiabilité 6
  - Sécurité 6
  - Surveillance 8
  - Interfaces de gestion 8
- Architecture 8
  - Stockage 9
  - Gestion des ressources de cluster 9
  - Infrastructures de traitement de données 10
  - Applications et programmes 10

Mise en route 12
- Étape 1 : Définir les prérequis 12
  - Inscrivez-vous à AWS 12
  - Création d'un compartiment Amazon S3 13
  - Création d'une paire de clés Amazon EC2 13
- Étape 2 : Lancer le cluster 13
  - Lancement de l'exemple de cluster 13
  - Récapitulatif des options rapides 14
- Étape 3 : Autoriser l'accès SSH 19
- Étape 4 : Exécution d'un script Hive pour traiter des données 21
  - Présentation des données et du script 21
  - Soumission du script Hive en tant qu'étape 22
  - Affichage des résultats 23
- Étape 5 : Nettoyage des ressources 24

Utilisation des blocs-notes EMR 25
- Considérations : 25
  - Exigences du cluster 25
  - Différences dans les capacités selon la version de cluster 26
  - Limites pour les ordinateurs portables connectés simultanément 27
  - Versions bloc-notes Jupyter et Python 27
  - 27
- Création d'un bloc-notes 27
- Utilisation des blocs-notes 29
  - Comprendre le statut de bloc-notes 29
  - Utilisation de l'éditeur de bloc-notes 30
  - Changement de clusters 31
  - Suppression des blocs-notes et des fichiers de bloc-notes 32
  - Partage de fichiers de bloc-notes 32
- Exécution Blocs-notes EMR Programmatique 33
  - Exemples de commande CLI 33
  - Script d’échantillon SDK Boto3 36
  - Script d’échantillon Rubis 38
- Emprunt d'identité pour Spark 40
  - Configuration de l'emprunt d'identité d'un utilisateur Spark 40

Utilisation du widget de surveillance de tâche Spark ... 41
Sécurité ... 42
Installation et utilisation des noyaux et des bibliothèques ... 43
Installation des noyaux et des bibliothèques Python sur un nœud maître de cluster ... 43
Utilisation de bibliothèques limitées aux bloc-notes ... 43
Association de référentiels Git à Blocs-notes EMR ... 44
Plan and Configure Clusters ... 50
Configure Cluster Location and Data Storage ... 50
Choose an AWS Region ... 50
Work with Storage and File Systems ... 52
Prepare Input Data ... 54
Configure an Output Location ... 63
Planification et configuration des nœuds maîtres ... 68
Applications et fonctionnalités prises en charge ... 69
Lancement d'un cluster EMR avec plusieurs nœuds maîtres ... 74
Considérations et bonnes pratiques ... 76
Clusters EMR sur AWS Outposts ... 77
Prérequis ... 77
Limites ... 77
Considérations relatives à la connectivité réseau ... 78
Création d'un cluster Amazon EMR sur AWS Outposts ... 78
Clusters EMR dans les Zones locales AWS ... 79
Types d'instance pris en charge ... 79
Création d'un cluster Amazon EMR dans des Zones locales ... 79
Configuration de Docker ... 80
Registres Docker ... 81
Configuration des registres Docker ... 81
Configuration de YARN pour accéder à Amazon ECR sur EMR 6.0.0 et versions antérieures ... 82
Utiliser le système de fichiers EMR (EMRFS) ... 84
Vue cohérente ... 84
Autorisation d'accès aux données EMRFS dans Amazon S3 ... 101
Spécification du chiffrement Amazon S3 en utilisant les propriétés EMRFS ... 102
Control Cluster Termination ... 110
Configuring a Cluster to Auto-Terminate or Continue ... 110
Using Termination Protection ... 111
Utilisation des AMI ... 116
À l'aide de l'AMI par défaut ... 117
Utilisation d'une image AMI personnalisée ... 118
Spécification de la taille du volume du périphérique racine Amazon EBS ... 123
Configure Cluster Software ... 125
Create Bootstrap Actions to Install Additional Software ... 125
Configuration du matériel et de la mise en réseau d'un cluster ... 129
Comprendre les types de nœuds ... 130
Configurer les instances EC2 ... 131
Configuration de la mise en réseau ... 137
Configurer des parcs ou groupes d'instances ... 148
Consignes et bonnes pratiques ... 161
Configure Cluster Logging and Debugging ... 166
Default Log Files ... 167
Archive Log Files to Amazon S3 ... 167
Enable the Debugging Tool ... 170
Debugging Option Information ... 171
Tag Clusters ... 171
Tag Restrictions ... 172
Tag Resources for Billing ... 173
Add Tags to a New Cluster ... 173
Adding Tags to an Existing Cluster ... 174

View Tags on a Cluster 175
Remove Tags from a Cluster 175
Drivers and Third-Party Application Integration 176
Use Business Intelligence Tools with Amazon EMR 176
Sécurité 177
Configurations de la sécurité 177
Protection des données 177
AWS Identity and Access Management avec Amazon EMR 177
Kerberos 178
Lake Formation 178
Secure Socket Shell (SSH) 178
Groupes de sécurité Amazon EC2 178
Mises à jour de l'AMI Amazon Linux par défaut 178
Utilisation de configurations de sécurité pour configurer la sécurité du cluster 179
Création d'une configuration de sécurité 179
Spécification d'une configuration de sécurité pour un cluster 196
Protection des données 197
Chiffrer les données au repos et en transit 197
IAM avec Amazon EMR 207
Audience 207
Authentification avec les identités 208
Gestion de l'accès à l'aide des stratégies 209
Comment Amazon EMR fonctionne avec IAM 211
Configure Service Roles for Amazon EMR 213
Exemples de stratégie basée sur l'identité 240
Authenticate to Cluster Nodes 255
Use an Amazon EC2 Key Pair for SSH Credentials 255
Use Kerberos Authentication 255
Intégration Amazon EMR avec AWS Lake Formation (Bêta) 282
Présentation conceptuelle de l'intégration Amazon EMR à Lake Formation 283
Applications et fonctionnalités prises en charge 288
Avant de commencer 289
Lancez un cluster Amazon EMR avec Lake Formation 296
Contrôle du trafic réseau avec des groupes de sécurité 301
Utilisation des groupes de sécurité gérés par Amazon EMR 303
Utilisation des groupes de sécurité supplémentaires 309
Spécification des groupes de sécurité 309
Groupes de sécurité pour Blocs-notes EMR 311
Utilisation de Block Public Access 312
Validation de la conformité 314
Résilience 315
Sécurité de l'infrastructure 315
Connexion à Amazon EMR à l'aide d'un point de terminaison de VPC d'interface 315
Manage Clusters 319
View and Monitor a Cluster 319
View Cluster Status and Details 319
Enhanced Step Debugging 325
Afficher l'historique de l'application 327
View Log Files 333
View Cluster Instances in Amazon EC2 337
Événements et métriques CloudWatch 338
View Cluster Application Metrics with Ganglia 364
Journalisation des appels d'API Amazon EMR dans AWS CloudTrail 365
Connexion au cluster 367
Connexion au nœud maître à l'aide de SSH 368
Affichage des interfaces web hébergées sur des clusters Amazon EMR 372
Arrêter un cluster 382

Mettre fin à un cluster à l'aide de la console ... 382
Arrêter un cluster à l'aide de l'AWS CLI ... 383
Arrêter un cluster à l'aide de l'API ... 383
Dimensionnement des ressources de cluster ... 384
Utilisation du dimensionnement automatique géré par EMR dans Amazon EMR ... 385
Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances ... 401
Redimensionnement manuel d'un cluster en cours d'exécution ... 410
Diminution de capacité des clusters ... 416
Cloning a Cluster Using the Console ... 417
Submit Work to a Cluster ... 418
Work with Steps Using the AWS CLI and Console ... 418
Submit Hadoop Jobs Interactively ... 423
Add More than 256 Steps to a Cluster ... 424
Automate Recurring Clusters with AWS Data Pipeline ... 425
Troubleshoot a Cluster ... 426
What Tools are Available for Troubleshooting? ... 426
Tools to Display Cluster Details ... 426
Tools to View Log Files ... 427
Tools to Monitor Cluster Performance ... 427
Viewing and Restarting Amazon EMR and Application Processes (Daemons) ... 428
Viewing Running Processes ... 428
Restarting Processes ... 428
Troubleshoot a Failed Cluster ... 429
Step 1: Gather Data About the Issue ... 430
Step 2: Check the Environment ... 430
Step 3: Look at the Last State Change ... 431
Step 4: Examine the Log Files ... 431
Step 5: Test the Cluster Step by Step ... 432
Résolution des problèmes de rapidité d'un cluster ... 433
Étape 1. Recueillir des données sur le problème ... 433
Étape 2. Vérifier l’environnement ... 434
Étape 3 Examiner les fichiers journaux ... 435
Étape 4. Vérifier la santé des groupes et des instances ... 436
Étape 5 : Vérifier les groupes suspendus ... 437
Étape 6 : Vérifier les paramètres de configuration ... 438
Étape 7. Examiner les données d’entrée ... 439
Common Errors in Amazon EMR ... 440
Input and Output Errors ... 440
Erreurs d'autorisations ... 442
Erreurs de ressource ... 443
Streaming Cluster Errors ... 450
Custom JAR Cluster Errors ... 451
Hive Cluster Errors ... 452
VPC Errors ... 453
AWS GovCloud (US-West) Errors ... 456
Other Issues ... 456
Dépannage d’un cluster Lake Formation (version bêta) ... 456
Expiration de session ... 456
Aucune autorisation pour l'utilisateur sur la table demandée ... 457
Insertion, création et modification de tables : non prises en charge en version bêta ... 457
Write Applications that Launch and Manage Clusters ... 458
End-to-End Amazon EMR Java Source Code Sample ... 458
Common Concepts for API Calls ... 461
Endpoints for Amazon EMR ... 461
Specifying Cluster Parameters in Amazon EMR ... 461
Availability Zones in Amazon EMR ... 462

How to Use Additional Files and Libraries in Amazon EMR Clusters 462
Use SDKs to Call Amazon EMR APIs 462
Using the AWS SDK for Java to Create an Amazon EMR Cluster 463
Glossaire AWS 465
..... cdlxvi

Qu'est-ce qu'Amazon EMR ?

Amazon EMR est une plate-forme de cluster gérée qui simplifie l'exécution des infrastructures de données massives, telles qu'Apache Hadoop et Apache Spark, sur AWS pour traiter et analyser de grandes quantités de données. Grâce à ces infrastructures et des projets open source connexes, tels qu'Apache Hive et Apache Pig, vous pouvez traiter des données à des fins d'analyse et pour des charges de travail d'aide à la décision. En outre, vous pouvez utiliser Amazon EMR pour transformer et transférer de grandes quantités de données dans et hors d'autres bases de données et magasins de données AWS, tels que Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB.

Si vous utilisez Amazon EMR pour la première fois, nous vous recommandons de commencer par lire les sections suivantes en plus de la présente section :

- Amazon EMR – Cette page de service fournit les points forts, la description détaillée et les informations de tarification de Amazon EMR.
- Démarrer : Analyse Big Data avec Amazon EMR (p. 12) – Ces didacticiels vous aide à démarrer rapidement avec Amazon EMR.

Dans cette section :

- Présentation d'Amazon EMR (p. 1)
- Avantages de l'utilisation d'Amazon EMR (p. 4)
- Présentation de l'architecture d'Amazon EMR (p. 8)

Présentation d'Amazon EMR

Cette rubrique fournit une présentation des clusters Amazon EMR, y compris de la façon de soumettre du travail à un cluster, de la manière dont ces données sont traitées et des différents états par lesquels le cluster passe au cours de ce traitement.

Dans cette rubrique

- Présentation des clusters et des nœuds (p. 1)
- Soumission de travail à un cluster (p. 2)
- Traitement des données (p. 2)
- Présentation du cycle de vie du cluster (p. 4)

Présentation des clusters et des nœuds

Le composant central de Amazon EMR est le cluster. Un cluster est un ensemble d'instances Amazon Elastic Compute Cloud (Amazon EC2). Chaque instance dans le cluster est appelée un nœud. Chaque nœud dispose d'un rôle dans le cluster, qu'on appelle le type de nœud. Amazon EMR installe également des composants logiciels différents sur chaque type de nœud, conférant ainsi à chaque nœud un rôle dans une application distribuée telle qu'Apache Hadoop.

Les types de nœud dans Amazon EMR sont les suivants :

- Nœud principal : nœud qui gère le cluster en exécutant des composants logiciels pour coordonner la distribution des données et des tâches entre d'autres nœuds en vue de leur traitement. Le nœud maître

effectue le suivi du statut des tâches et surveille l'état du cluster. Chaque cluster a un nœud maître ; il est possible de créer un cluster à nœud unique avec seulement le nœud maître.

- Nœud de noyau : nœud doté de composants logiciels qui exécutent les tâches et stockent les données dans le système de fichiers distribué Hadoop (HDFS) sur votre cluster. Les clusters à plusieurs nœuds ont au moins un nœud principal.
- Nœud de tâche : nœud doté de composants logiciels qui exécutent uniquement des tâches et ne stockent pas les données dans HDFS. Les nœuds de tâches sont facultatifs.

Le schéma suivant représente un cluster avec un seul nœud principal et quatre nœuds de noyau.

Soumission de travail à un cluster

Lorsque vous exécutez un cluster sur Amazon EMR, vous avez plusieurs options quant à la façon de spécifier le travail qui doit être effectué.

- Fournissez la définition complète du travail à effectuer dans des fonctions que vous spécifiez en tant qu'étapes lorsque vous créez un cluster. Cette solution est privilégiée pour les clusters qui traitent une quantité déterminée de données, puis sont arrêtés une fois le traitement terminé.
- Créez un cluster de longue durée et utilisez la console Amazon EMR, l'API Amazon EMR ou l'AWS CLI pour soumettre des étapes, lesquelles peuvent contenir un travail ou plusieurs travaux. Pour plus d'informations, consultez Submit Work to a Cluster (p. 418).
- Créez un cluster, connectez-vous au nœud maître et aux autres nœuds si nécessaire à l'aide de SSH, puis utilisez les interfaces que les applications installées fournissent pour effectuer des tâches et soumettre des requêtes, soit par l'intermédiaire de scripts soit de manière interactive. Pour plus d'informations, consultez le manuel Amazon EMR Guide de version.

Traitement des données

Lorsque vous lancez votre cluster, vous choisissez les infrastructures et les applications à installer pour répondre à vos besoins de traitement des données. Pour traiter les données de votre cluster Amazon

EMR, vous pouvez soumettre des travaux ou des requêtes directement aux applications installées, ou vous pouvez exécuter des étapes dans le cluster.

Soumission de travaux directement aux applications

Vous pouvez soumettre des travaux et interagir directement avec le logiciel qui est installé dans votre cluster Amazon EMR. Pour cela, vous vous connectez généralement au nœud maître via une connexion sécurisée et accédez aux interfaces et outils qui sont disponibles pour le logiciel qui s'exécute directement sur votre cluster. Pour plus d'informations, consultez Connexion au cluster (p. 367).

Exécution d'étapes pour traiter des données

Vous pouvez soumettre une ou plusieurs étapes ordonnées à un cluster Amazon EMR. Chaque étape est une unité de travail qui contient des instructions de manipulation des données qui doivent être traitées par le logiciel installé sur le cluster.

Voici un exemple de processus à quatre étapes :

1. Envoi d'un ensemble de données d'entrée à traiter.
2. Traitement des données de sortie de la première étape à l'aide d'un programme Pig.
3. Traitement d'un second ensemble de données d'entrée à l'aide d'un programme Hive.
4. Écriture d'un ensemble de données de sortie.

En règle générale, lorsque vous traitez des données dans Amazon EMR, l'entrée correspond à des données stockées sous forme de fichiers dans votre système de fichiers sous-jacent choisi, tel que Amazon S3 ou HDFS. Ces données passent d'une étape à l'autre dans la séquence de traitement. L'étape finale écrit les données de sortie dans un emplacement spécifié, tel qu'un compartiment Amazon S3.

Les étapes sont exécutées dans l'ordre suivant :

1. Une demande est soumise pour commencer le traitement des étapes.
2. L'état de toutes les étapes est défini sur EN SUSPENS.
3. Lorsque la première étape de la séquence commence, son état passe à EN COURS D'EXÉCUTION. Les autres étapes restent à l'état EN SUSPENS.
4. Une fois la première étape terminée, son état devient TERMINÉ.
5. L'étape suivante de la séquence commence et son état passe à EN COURS D'EXÉCUTION. Une fois terminée, son état devient TERMINÉ.
6. Ce modèle se répète pour chaque étape jusqu'à ce qu'elles soient toutes terminées, puis le traitement se termine.

Le schéma suivant représente la séquence d'étapes et le changement d'état des étapes au fur et à mesure de leur traitement.

Si une étape échoue au cours du traitement, son état devient TERMINATED_WITH_ERRORS. Vous pouvez déterminer ce qui se passe ensuite pour chaque étape. Par défaut, les étapes restantes de la séquence sont définies sur CANCELLED (ANNULÉE) et ne sont pas exécutées. Vous pouvez également choisir d'ignorer l'échec et d'autoriser l'exécution des étapes restantes, ou d'arrêter le cluster immédiatement.

Le schéma suivant représente la séquence des étapes et le changement d'état par défaut lorsqu'une étape échoue pendant le traitement.

Présentation du cycle de vie du cluster

Un cluster Amazon EMR réussi suit ce processus :

1. Amazon EMR alloue tout d'abord des instances EC2 dans le cluster pour chaque instance en fonction de vos spécifications. Pour plus d'informations, consultez Configuration du matériel et de la mise en réseau d'un cluster (p. 129). Pour toutes les instances, Amazon EMR utilise l'AMI par défaut pour Amazon EMR ou une AMI Amazon Linux personnalisée que vous spécifiez. Pour plus d'informations, consultez Utilisation d'une image AMI personnalisée (p. 118). Au cours de cette phase, l'état du cluster est `STARTING`.
2. Amazon EMR exécute les actions d'amorçage que vous spécifiez sur chaque instance. Vous pouvez utiliser des actions d'amorçage pour installer des applications personnalisées et exécuter les personnalisations dont vous avez besoin. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125). Au cours de cette phase, l'état du cluster est `BOOTSTRAPPING`.
3. Amazon EMR installe les applications natives que vous spécifiez lorsque vous créez le cluster, telles que Hive, Hadoop, Spark, etc.
4. Lorsque les actions d'amorçage sont terminées et que les applications natives sont installées, l'état du cluster est `RUNNING`. À ce stade, vous pouvez vous connecter aux instances de cluster, et le cluster exécute de façon séquentielle toutes les étapes que vous avez spécifiées lorsque vous avez créé le cluster. Vous pouvez ajouter des étapes supplémentaires, qui s'exécuteront lorsque les étapes précédentes seront terminées. Pour plus d'informations, consultez Work with Steps Using the AWS CLI and Console (p. 418).
5. Une fois que les étapes ont été exécutées avec succès, le cluster passe à l'état `WAITING`. Si un cluster est configuré pour s'arrêter automatiquement après la dernière étape, il passe à l'état `SHUTTING_DOWN`.
6. Lorsque toutes les instances sont arrêtées, le cluster passe à l'état `COMPLETED`.

En cas de défaillance au cours du cycle de vie du cluster, Amazon EMR arrête le cluster et toutes ses instances, sauf si vous activez la protection contre l'arrêt. Si un cluster s'arrête en raison d'une défaillance, toutes les données stockées sur le cluster sont supprimées, et son état devient `FAILED`. Si vous avez activé la protection contre l'arrêt, vous pouvez récupérer les données à partir de votre cluster, puis supprimer la protection contre l'arrêt et arrêter le cluster. Pour plus d'informations, consultez Using Termination Protection (p. 111).

Avantages de l'utilisation d'Amazon EMR

Il existe de nombreux avantages à l'utilisation d'Amazon EMR. Cette section fournit une présentation de ces avantages et des liens vers des informations supplémentaires qui vous aideront à approfondir le sujet.

Rubriques

- Economies sur les coûts (p. 5)
- Intégration à AWS (p. 5)
- Déploiement (p. 5)
- Evolutivité et flexibilité (p. 6)
- Fiabilité (p. 6)

- Sécurité (p. 6)
- Surveillance (p. 8)
- Interfaces de gestion (p. 8)

Economies sur les coûts

La tarification d'Amazon EMR dépend du type d'instance et du nombre d'instances EC2 que vous déployez, ainsi que de la région dans laquelle vous lancez votre cluster. La tarification à la demande offre des taux horaires faibles, mais vous pouvez réduire encore le coût en achetant des instances réservées ou en faisant une offre sur des instances Spot. Les instances Spot permettent des économies importantes et peuvent même parfois ne représenter qu'un dixième de la tarification à la demande.

> Note
>
> Si vous utilisez Amazon S3, Amazon Kinesis ou DynamoDB avec votre cluster EMR, des frais supplémentaires s'appliquent pour les services qui sont facturés séparément de votre utilisation de Amazon EMR.

Pour de plus amples informations sur les options et les détails de tarification, veuillez consulter Tarification de Amazon EMR.

Intégration à AWS

Amazon EMR s'intègre à d'autres services AWS pour fournir des capacités et des fonctionnalités de mise en réseau, stockage, sécurité, etc., pour votre cluster. La liste suivante fournit plusieurs exemples de cette intégration :

- Amazon EC2 pour les instances qui incluent les nœuds du cluster
- Amazon Virtual Private Cloud (Amazon VPC) pour configurer le réseau virtuel dans lequel vous lancez vos instances
- Amazon S3 pour stocker les données d'entrée et sortie
- Amazon CloudWatch pour surveiller les performances de cluster et configurer des alarmes
- AWS Identity and Access Management (IAM) pour configurer les autorisations
- AWS CloudTrail pour effectuer l'audit des demandes effectuées auprès du service
- AWS Data Pipeline pour planifier et démarrer vos clusters

Déploiement

Votre cluster EMR se compose d'instances EC2, qui effectuent le travail que vous soumettez à votre cluster. Lorsque vous lancez votre cluster, Amazon EMR configure les instances avec les applications que vous choisissez, telles qu'Apache Hadoop ou Spark. Choisissez le type et la taille d'instance qui conviennent le mieux aux besoins de traitement pour votre cluster : traitement par lots, requêtes à faible latence, streaming de données ou stockage de données volumineuses. Pour plus d'informations sur les types d'instances disponibles pour Amazon EMR, consultez Configuration du matériel et de la mise en réseau d'un cluster (p. 129).

Amazon EMR offre diverses façons de configurer des logiciels sur votre cluster. Par exemple, vous pouvez installer une version Amazon EMR avec un ensemble choisi d'applications qui peut inclure des infrastructures polyvalentes, telles que Hadoop, et des applications, telles que Hive, Pig ou Spark. Vous pouvez également installer une distribution MapR. Amazon EMR utilise Amazon Linux, si bien que vous pouvez également installer des logiciels sur votre cluster manuellement en utilisant le gestionnaire de package yum ou à partir de la source. Pour plus d'informations, consultez Configure Cluster Software (p. 125).

Evolutivité et flexibilité

Amazon EMR offre une grande flexibilité pour augmenter ou réduire la taille de votre cluster lorsque vos besoins informatiques évoluent. Vous pouvez redimensionner votre cluster pour ajouter des instances pour les charges de travail des périodes de pointe et supprimer des instances pour contrôler les coûts en dehors des périodes de pointe. Pour plus d'informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution (p. 410).

Amazon EMR offre également la possibilité d'exécuter plusieurs groupes d'instances pour vous permettre d'utiliser les instances à la demande dans un groupe afin de garantir la puissance de traitement, et les instances Spot dans un autre groupe afin de terminer plus rapidement et à meilleur coût vos travaux. Vous pouvez également combiner différents types d'instance pour tirer profit de meilleurs prix pour un type d'instance Spot par rapport à un autre. Pour plus d'informations, consultez Quand faut-il utiliser des instances Spot ? (p. 162).

De plus, Amazon EMR offre la possibilité d'utiliser plusieurs systèmes de fichiers pour vos données d'entrée, de sortie et intermédiaires. Par exemple, vous pouvez choisir le système de fichiers distribué Hadoop (HDFS) qui s'exécute sur les nœuds maître et principaux de votre cluster pour traiter les données que vous n'avez pas besoin de stocker au-delà du cycle de vie de votre cluster. Vous pouvez choisir le système de fichiers EMR (EMRFS) pour utiliser Amazon S3 comme une couche de données pour les applications qui s'exécutent sur votre cluster. Vous pouvez ainsi séparer les calculs et le stockage, et conserver les données en dehors du cycle de vie de votre cluster. EMRFS offre l'avantage supplémentaire de vous permettre de monter ou descendre en puissance, indépendamment en fonction de vos besoins de calcul et de stockage. Vous pouvez ajuster vos besoins de calcul en redimensionnant votre cluster et vous pouvez ajuster vos besoins de stockage en utilisant Amazon S3. Pour plus d'informations, consultez Work with Storage and File Systems (p. 52).

Fiabilité

Amazon EMR surveille les nœuds de votre cluster, et résilie automatiquement une instance et la remplace en cas de défaillance.

Amazon EMR fournit des options de configuration qui contrôlent la manière dont votre cluster est arrêté, automatiquement ou manuellement. Si vous configurez votre cluster pour qu'il s'arrête automatiquement, il est arrêté une fois toutes les étapes terminées. On parle alors de cluster transitoire. Toutefois, vous pouvez configurer le cluster pour qu'il continue à s'exécuter après la fin du traitement, afin que vous puissiez choisir de l'arrêter manuellement lorsque vous n'en avez plus besoin. Ou, vous pouvez créer un cluster, interagir directement avec les applications installées, puis arrêter manuellement le cluster lorsque vous n'en avez plus besoin. Les clusters de ces exemples sont appelés clusters de longue durée.

De plus, vous pouvez configurer une protection contre l'arrêt pour empêcher les instances de votre cluster d'être mises hors service en raison d'erreurs ou de problèmes au cours du traitement. Lorsque la protection de la résiliation est activée, vous pouvez récupérer les données à partir des instances avant leur résiliation. Les paramètres par défaut de ces options varient selon que vous lancez votre cluster à l'aide de la console, de l'interface de ligne de commande ou de l'API. Pour plus d'informations, consultez Using Termination Protection (p. 111).

Sécurité

Amazon EMR exploite d'autres services AWS, tels que IAM et Amazon VPC, ainsi que des fonctionnalités telles que les paires de clés Amazon EC2, afin de vous aider à sécuriser vos clusters et vos données.

IAM

Amazon EMR s'intègre à IAM pour gérer les autorisations. Vous définissez des autorisations à l'aide de stratégies IAM, que vous attachez à des utilisateurs IAM ou à des groupes IAM. Les autorisations que vous

définissez dans la stratégie déterminent les actions que les utilisateurs ou les membres du groupe peuvent effectuer et les ressources auxquelles ils peuvent accéder. Pour de plus amples informations, veuillez consulter Comment Amazon EMR fonctionne avec IAM (p. 211).

De plus, Amazon EMR utilise les rôles IAM pour le service Amazon EMR lui-même et le profil d'instance EC2 pour les instances. Ces rôles accordent au service et aux instances l'autorisation d'accéder à d'autres services AWS pour votre compte. Il existe un rôle par défaut pour le service Amazon EMR et un rôle par défaut pour le profil d'instance EC2. Les rôles par défaut utilisent les stratégies gérées AWS, qui sont créées pour vous automatiquement la première fois que vous lancez un cluster EMR à partir de la console et que vous choisissez les autorisations par défaut. Vous pouvez également créer les rôles IAM par défaut à partir de l'AWS CLI. Si vous voulez gérer les autorisations à la place d'AWS, vous pouvez choisir des rôles personnalisés pour le service et le profil d'instance. Pour plus d'informations, consultez Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).

Groupes de sécurité

Amazon EMR utilise des groupes de sécurité pour contrôler le trafic entrant et sortant de vos instances EC2. Lorsque vous lancez votre cluster, Amazon EMR utilise un groupe de sécurité pour votre instance principale ainsi qu'un groupe de sécurité à partager avec vos instances de noyau/de tâches. Amazon EMR configure les règles de groupe de sécurité pour assurer la communication entre les instances dans le cluster. Si vous le souhaitez, vous pouvez configurer des groupes de sécurité supplémentaires et les affecter à vos instances maîtres et principales/de tâches pour obtenir des règles plus avancées. Pour plus d'informations, consultez Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Chiffrement

Amazon EMR prend en charge le chiffrement facultatif côté serveur et côté client de Amazon S3 avec EMRFS pour favoriser la protection des données que vous stockez dans Amazon S3. Avec le chiffrement côté serveur, Amazon S3 chiffre vos données une fois que vous les avez chargées vers le serveur.

Avec le chiffrement côté client, le processus de chiffrement et de déchiffrement se produit dans le client EMRFS, sur votre cluster EMR. Vous gérez la clé principale pour le chiffrement côté client à l'aide de AWS Key Management Service (AWS KMS) ou de votre propre système de gestion de clés.

Pour plus d'informations, consultez Chiffrement des données Amazon S3 avec EMRFS dans le Amazon EMR Guide de version.

Amazon VPC

Amazon EMR prend en charge le lancement des clusters dans un Virtual Private Cloud (VPC) dans Amazon VPC. Un VPC est un réseau virtuel isolé dans AWS qui offre la possibilité de contrôler les aspects avancés de la configuration du réseau et de son accès. Pour plus d'informations, consultez Configuration de la mise en réseau (p. 137).

AWS CloudTrail

Amazon EMR s'intègre à CloudTrail pour consigner des informations sur les demandes effectuées par ou au nom de votre compte AWS. Avec ces informations, vous pouvez obtenir un suivi des personnes qui accèdent à votre cluster, des heures où cela se produit et de l'adresse IP à partir de laquelle elles effectuent la demande. Pour plus d'informations, consultez Journalisation des appels d'API Amazon EMR dans AWS CloudTrail (p. 365).

Paires de clés Amazon EC2

Vous pouvez surveiller votre cluster et interagir avec lui en créant une connexion sécurisée entre votre ordinateur distant et le nœud maître. Vous pouvez utiliser le protocole réseau SSH (Secure Shell) pour

cette connexion ou utiliser Kerberos pour l'authentification. Si vous utilisez SSH, une paire de clés Amazon EC2 est obligatoire. Pour plus d'informations, consultez Use an Amazon EC2 Key Pair for SSH Credentials (p. 255).

Surveillance

Vous pouvez utiliser les interfaces de gestion Amazon EMR et les fichiers de journaux pour dépanner vos problèmes de cluster, comme des erreurs ou des échecs. Amazon EMR propose la capacité d'archiver des fichiers de journaux dans Amazon S3 afin que vous puissiez stocker des journaux et dépanner des problèmes, même après la résiliation de votre cluster. Amazon EMR fournit également un outil de débogage facultatif dans la console Amazon EMR pour parcourir les fichiers de journaux selon les étapes, les travaux et les tâches. Pour plus d'informations, consultez Configure Cluster Logging and Debugging (p. 166).

Amazon EMR s'intègre à CloudWatch pour effectuer le suivi des métriques de performances pour le cluster et des travaux au sein du cluster. Vous pouvez configurer des alarmes sur la base de diverses métriques, telles que le fait que le cluster soit ou non inactif ou le pourcentage de stockage utilisé. Pour plus d'informations, consultez Surveillance des métriques avec CloudWatch (p. 348).

Interfaces de gestion

Il existe plusieurs manières d'interagir avec Amazon EMR :

- Console — Interface utilisateur graphique qui permet de lancer et gérer des clusters. Elle vous permet de remplir des formulaires web afin de préciser les détails relatifs aux clusters à lancer, de consulter les informations relatives aux clusters en cours, de déboguer et d'arrêter les clusters. Cette console constitue le moyen le plus simple de faire ses premiers pas avec Amazon EMR ; aucune connaissance en programmation n'est requise. La console est disponible en ligne à https://console.aws.amazon.com/elasticmapreduce/home.
- AWS Command Line Interface (AWS CLI) — Application cliente que vous exécutez sur votre machine locale pour vous connecter à Amazon EMR et pour créer et gérer des clusters. L'AWS CLI contient un ensemble de commandes, doté de nombreuses fonctions, spécifique à Amazon EMR. Elle vous permet d'écrire des scripts pour automatiser le lancement et la gestion des clusters. Si vous préférez travailler à partir d'une ligne de commande, l'utilisation de l'AWS CLI est la meilleure option. Pour plus d'informations, consultez Amazon EMR dans le AWS CLI Command Reference.
- Kit de développement logiciel (SDK) — Les kits SDK fournissent des fonctions pour appeler Amazon EMR afin de créer et gérer des clusters. Ils vous permettent d'écrire des applications pour automatiser la création et la gestion des clusters. L'utilisation du kit SDK est la meilleure manière de prolonger ou de personnaliser la fonctionnalité d'Amazon EMR. Amazon EMR est actuellement disponible dans les kits SDK suivants : Go, Java, .NET (C# et VB.NET), Node.js, PHP, Python et Ruby. Pour de plus amples informations sur ces SDK, veuillez consulter les rubriques Outils pour AWS et Exemples de codes et bibliothèques Amazon EMR.
- API de service web — Interface de bas niveau qui vous permet d'appeler le service web directement à l'aide de JSON. Cette API est l'option la plus adaptée pour créer un kit SDK personnalisé qui appelle Amazon EMR. Pour en savoir plus, consultez le Amazon EMR API Reference.

Présentation de l'architecture d'Amazon EMR

L'architecture du service Amazon EMR se compose de plusieurs couches, chacune fournissant certaines fonctions et fonctionnalités au cluster. Cette section fournit une présentation de ces couches et de leurs composants.

Dans cette rubrique

- Stockage (p. 9)

- Gestion des ressources de cluster (p. 9)
- Infrastructures de traitement de données (p. 10)
- Applications et programmes (p. 10)

Stockage

La couche de stockage inclut les différents systèmes de fichiers qui sont utilisés avec votre cluster. Il existe plusieurs types d'options de stockage, décrits ci-dessous.

Système de fichiers distribué Hadoop (HDFS)

Le système de fichiers distribué Hadoop (HDFS) est un système de fichiers évolutif, distribué pour Hadoop. Le système HDFS répartit les données qu'il stocke entre les instances dans le cluster, stockant plusieurs copies des données sur différentes instances pour garantir qu'aucune donnée n'est perdue en cas de défaillance d'une instance individuelle. HDFS est un stockage éphémère qui est récupéré lorsque vous mettez fin à un cluster. HDFS est utile pour la mise en cache des résultats intermédiaires au cours du traitement MapReduce ou pour les charges de travail qui ont des E/S aléatoires importantes.

Pour plus d'informations, consultez le Guide de l'utilisateur HDFS sur le site web d'Apache Hadoop.

Système de fichiers EMR (EMRFS)

Avec le système de fichiers EMR (EMRFS), Amazon EMR étend Hadoop pour ajouter la possibilité d'accéder directement aux données stockées dans Amazon S3 comme s'il s'agissait d'un système de fichiers de type HDFS. Vous pouvez utiliser HDFS ou Amazon S3 en tant que système de fichiers dans votre cluster. Le plus souvent, Amazon S3 est utilisé pour stocker les données d'entrée et sortie et les résultats intermédiaires sont stockés dans HDFS.

Système de fichiers local

Le système de fichiers local fait référence à un disque connecté localement. Lorsque vous créez un cluster Hadoop, chaque nœud est créé à partir d'une instance Amazon EC2 qui est associée à un bloc préconfiguré de stockage sur disque pré-attaché appelé stockage d'instance. Les données des volumes de stockage d'instance sont conservées uniquement pendant le cycle de durée de vie de leur instance Amazon EC2.

Gestion des ressources de cluster

La couche de gestion des ressources est responsable de la gestion des ressources du cluster et de la planification des travaux de traitement des données.

Par défaut, Amazon EMR utilise YARN (Yet Another Resource Negotiator), un composant introduit dans Apache Hadoop 2.0 pour gérer de manière centralisée les ressources de cluster pour plusieurs infrastructures de traitement de données. Cependant, il existe d'autres cadres et applications qui sont proposés par Amazon EMR qui n'utilisent pas YARN comme gestionnaire de ressources. Amazon EMR dispose également d'un agent sur chaque nœud pour administrer les composants YARN, maintenir la santé du cluster et communiquer avec Amazon EMR.

Les instances Spot étant souvent utilisées pour exécuter des nœuds de tâche, Amazon EMR dispose d'une fonctionnalité par défaut pour la planification de tâches YARN de sorte que les tâches en cours d'exécution n'échouent pas lorsque des nœuds de tâche en cours d'exécution sur des instances Spot sont résiliées. Amazon EMR s'en assure en n'autorisant les processus maîtres des applications à s'exécuter uniquement sur des nœuds principaux. Le processus maître de l'application contrôle l'exécution des tâches et doit rester actif pendant toute la durée de vie de la tâche.

Amazon EMR version 5.19.0 et ultérieures utilise la fonction des étiquettes de nœud YARN intégrés pour y parvenir. (Les versions antérieures utilisaient un correctif de code). Les propriétés des classifications de configuration `yarn-site` et `capacity-scheduler` sont configurées par défaut de sorte que le planificateur de YARN et le planificateur d'équité tirent parti des étiquettes de nœud. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de sorte que les masters de l'application ne soient planifiés que sur des nœud avec l'étiquette CORE. La modification manuelle de propriétés apparentées dans les classifications de configuration yarn-site et capacity-scheduler ou directement dans les fichiers XML associés pourrait rompre cette fonctionnalité ou la modifier.

Infrastructures de traitement de données

La couche des infrastructures de traitement des données est le moteur utilisé pour traiter et analyser les données. Il existe de nombreuses infrastructures disponibles qui s'exécutent sur YARN ou qui possèdent leur propre gestion des ressources. Des infrastructures différentes sont disponibles pour les différents types de traitement requis, tels que les traitements par lots, interactif, en mémoire, de streaming, etc. L'infrastructure que vous choisissez dépend de votre cas d'utilisation. Elle affecte les langues et les interfaces disponibles à partir de la couche d'application, qui est la couche utilisée pour interagir avec les données à traiter. Les infrastructures de traitement principales disponibles pour Amazon EMR sont Hadoop MapReduce et Spark.

Hadoop MapReduce

Hadoop MapReduce est un modèle de programmation open source pour l'informatique distribuée. Il simplifie le processus d'écriture d'applications distribuées en parallèle en gérant l'ensemble de la logique, alors que vous fournissez les fonctions Map et Reduce. La fonction Map mappe les données aux ensembles de paires clé-valeur nommées résultats intermédiaires. La fonction Reduce, quant à elle, combine les résultats intermédiaires et leur applique d'autres algorithmes afin de générer la sortie finale. Il existe plusieurs infrastructures disponibles pour MapReduce, telles que Hive, qui génèrent automatiquement des programmes Map et Reduce.

Pour plus d'informations, consultez Comment les opérations de mappage et de réduction sont effectuées sur le site web Wiki d'Apache Hadoop.

Apache Spark

Spark est une infrastructure de cluster et un modèle de programmation pour le traitement de charges de travail de données volumineuses. A l'instar de Hadoop MapReduce, Spark est un système de traitement distribué en open source, mais il utilise des graphes acycliques dirigés comme plans d'exécution et tire parti de la mise en cache en mémoire pour les ensembles de données. Lorsque vous exécutez Spark sur Amazon EMR, vous pouvez utiliser EMRFS pour accéder directement à vos données dans Amazon S3. Spark prend en charge plusieurs modules de requête interactifs comme SparkSQL.

Pour plus d'informations, consultez Configuration Spark sur des clusters Amazon EMR dans le Amazon EMR Guide de version.

Applications et programmes

Amazon EMR prend en charge de nombreuses applications, dont Hive, Pig, et la bibliothèque Spark Streaming, pour fournir des fonctionnalités telles que l'utilisation de langages de niveau supérieur pour créer des charges de travail de traitement, l'exploitation d'algorithmes d'apprentissage automatique, l'élaboration d'applications de traitement de flux et la création d'entrepôts de données. En outre, Amazon EMR prend également en charge des projets open source qui possèdent leurs propres fonctionnalités de gestion de cluster au lieu d'utiliser YARN.

Vous utilisez diverses bibliothèques et divers langages pour interagir avec les applications que vous exécutez dans Amazon EMR. Par exemple, vous pouvez utiliser Java, Hive ou Pig avec MapReduce ou Spark Streaming, Spark SQL, MLlib et GraphX avec Spark.

Pour plus d'informations, consultez le manuel Amazon EMR Guide de version.

Démarrer : Analyse Big Data avec Amazon EMR

Ce didacticiel vous guide tout au long du processus de création d'un exemple de cluster Amazon EMR à l'aide des options Quick Create (Création rapide) dans la AWS Management Console. Une fois que vous avez créé le cluster, vous soumettez un script Hive en tant qu'étape pour traiter des exemples de données stockés dans Amazon Simple Storage Service (Amazon S3).

Ce didacticiel n'est pas destiné aux environnements de production et il ne couvre pas les options de configuration en profondeur. Il est destiné à vous aider à configurer aussi vite que possible un cluster à des fins d'évaluation. Si vous avez des questions ou que vous êtes bloqué, contactez l'équipe Amazon EMR en écrivant dans notre Forum de discussion.

L'exemple de cluster que vous créez s'exécute dans un environnement en direct. Les charges s'accumulent pour les instances de cluster à la vitesse par seconde pour les tarifs Amazon EMR. Pour plus d'informations, consultez Tarification Amazon EMR. Ces frais varient selon la région. Le coût doit être minime, car le cluster doit s'exécuter pendant moins d'une heure après que le cluster soit mis en service.

Des frais peuvent également s'accumuler pour le stockage de fichiers de sortie de requêtes que vous stockez dans Amazon S3 dans le cadre du didacticiel. Le fichier est petit, donc les frais devraient être minimaux. De plus, s'il s'agit de la première année que vous utilisez AWS, tout ou partie de vos frais pour Amazon S3 peuvent être supprimés si vous restez dans les limites d'utilisation de l'offre gratuite AWS. Pour plus d'informations, consultez Tarification Amazon S3 et Offre gratuite AWS.

Étapes de ce didacticiel

- Étape 1 : Configuration des conditions préalables pour votre exemple de cluster (p. 12)
- Étape 2 : Lancement de votre exemple de cluster Amazon EMR (p. 13)
- Étape 3 : Autoriser les connexions SSH vers le cluster à partir de votre client (p. 19)
- Étape 4 : Traitez des données en exécutant le script Hive en tant qu'étape (p. 21)
- Étape 5 : Résilier le cluster et supprimer le compartiment (p. 24)

Étape 1 : Configuration des conditions préalables pour votre exemple de cluster

Avant de commencer la configuration de votre cluster Amazon EMR, veillez à satisfaire les conditions préalables exposées dans cette rubrique.

Inscrivez-vous à AWS

Si vous n'avez pas de compte AWS, suivez la procédure suivante pour en créer un.

Pour s'inscrire sur AWS

1. Ouvrez https://aws.amazon.com/, puis choisissez Create an AWS Account.
2. Suivez les instructions en ligne.

Création d'un compartiment Amazon S3

Dans ce didacticiel, vous spécifiez un compartiment Amazon S3 et un dossier pour stocker les données de sortie à partir d'une requête Hive. Le didacticiel utilise l'emplacement de journal par défaut, mais vous pouvez également spécifier un emplacement personnalisé si vous préférez. En raison des exigences de Hadoop, les noms de compartiment et de dossier que vous utilisez avec Amazon EMR disposent des limitations suivantes :

- Ils doivent contenir uniquement des lettres, des chiffres, des points (.) et des traits d'union (-).
- Ils ne peuvent pas se terminer par des chiffres.

Si vous pouvez déjà accéder à un dossier qui répond à ces exigences, vous pouvez l'utiliser pour ce tutoriel. Le dossier de sortie doit être vide. Une autre exigence dont il faut se souvenir est que les noms des compartiments doivent être uniques sur tous les comptes AWS.

Pour plus d'informations sur la création d'un compartiment, consultez Création d'un compartiment dans le Amazon Simple Storage Service Guide de mise en route. Une fois que vous avez créé le compartiment, choisissez-le dans la liste, puis choisissez Create folder (Créer un dossier), remplacez New folder (Nouveau dossier) par un nom qui répond aux exigences, puis choisissez Save (Enregistrer).

Le nom de dossier et de compartiment utilisé plus tard dans le didacticiel est `s3 ://mybucket/MyHiveQueryResults`.

Création d'une paire de clés Amazon EC2

Vous devez disposer d'une paire de clés Amazon Elastic Compute Cloud (Amazon EC2) pour vous connecter aux nœuds de votre cluster via un canal sécurisé à l'aide du protocole SSH (Secure Shell). Si vous avez déjà une paire de clés que vous souhaitez utiliser, vous pouvez ignorer cette étape. Si vous n'avez pas de paire de clés, suivez l'une des procédures suivantes en fonction de votre système d'exploitation.

- Création de votre paire de clés à l'aide d'Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Windows.
- Création de votre paire de clés à l'aide d'Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Utilisez également cette procédure pour Mac OS.

Étape 2 : Lancement de votre exemple de cluster Amazon EMR

Dans cette étape, vous lancez votre exemple de cluster à l'aide des Quick Options (Options rapides) dans la console Amazon EMR, en conservant les valeurs par défaut de la plupart des options. Pour en savoir plus sur ces options, consultez Récapitulatif des options rapides (p. 14) après la procédure. Vous pouvez également sélectionner Go to advanced options (Accéder aux options avancées) pour explorer les options de configuration supplémentaires disponibles pour un cluster. Avant de créer votre cluster pour ce didacticiel, veillez à satisfaire les exigences exposées dans Étape 1 : Configuration des conditions préalables pour votre exemple de cluster (p. 12).

Lancement de l'exemple de cluster

Pour lancer l'exemple de cluster Amazon EMR

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.

2. Choisissez Créer un cluster.
3. Dans la page Create Cluster - Quick Options (Créer un cluster - Options rapides), acceptez les valeurs par défaut, sauf pour les champs suivants :
   - Saisissez un Cluster name (Nom de cluster) qui vous aidera à identifier le cluster, par exemple, `Mon premier cluster EMR`.
   - Sous Security and access (Sécurité et accès), choisissez la paire de clés EC2 que vous avez créée dans Création d'une paire de clés Amazon EC2 (p. 13).
4. Choisissez Créer un cluster.

La page d'état du cluster avec le cluster Summary (Récapitulatif) s'affiche. Vous pouvez utiliser cette page pour surveiller la progression de la création du cluster et afficher les détails sur le statut du cluster. Les éléments sur la page de statut sont mis à jour une fois les tâches de création de cluster terminées. Vous devrez peut-être choisir l'icône d'actualisation sur la droite ou actualisez votre navigateur pour recevoir des mises à jour.

Sous Network and hardware (Réseau et matériel), recherchez le statut d'instance Master (Principal) et Core (Noyau). Le statut passe de Provisioning (Mise en service) à Amorçage à Waiting (Attente) au cours du processus de création du cluster. Pour plus d'informations, consultez Présentation du cycle de vie du cluster (p. 4).

Dès que vous consultez les liens pour Security groups for Master (Groupes de sécurité pour le principal) et Security Groups for Core & Task (Groupes de sécurité pour le noyau et la tâche), vous pouvez passer à l'étape suivante. Vous pourriez vouloir attendre que le cluster démarre avec succès et soit en état d'Waiting (Attente).

Pour plus d'informations sur la lecture d'un résumé de cluster, consultez View Cluster Status and Details (p. 319).

Récapitulatif des options rapides

Le tableau suivant décrit les champs et les valeurs par défaut utilisés lors du lancement d'un cluster à l'aide de la page Quick cluster configuration (Configuration de cluster rapide) dans la console Amazon EMR.

Champ de la console	Valeur par défaut	Description
Nom du cluster	My cluster (Mon cluster)	Le nom de cluster est un nom descriptif facultatif pour votre cluster, qui n'a pas besoin d'être unique.
Journalisation	Activer	Lorsque la journalisation est activée, Amazon EMR écrit les données de journal détaillées dans le dossier Amazon S3 spécifié. Journalisation ne peut être activée que lorsque vous créez le cluster et le paramètre ne peut pas être modifié ultérieurement. Un compartiment Amazon S3 par défaut est spécifié. Vous pouvez éventuellement spécifier le votre. Pour plus d'informations, consultez View

Champ de la console	Valeur par défaut	Description
		Log Files Archived to Amazon S3 (p. 335).
S3 folder (Dossier S3)	s3://aws-logs-***numéro_compte-région***/elasticmapreduce/	Cette option spécifie le chemin d'accès à un dossier dans un compartiment Amazon S3 dans lequel vous voulez qu'Amazon EMR écrive les données de journal. Si le dossier par défaut du chemin d'accès spécifié n'existe pas dans le compartiment, il est créé pour vous. Vous pouvez spécifier un autre dossier en tapant ou en naviguant vers un dossier Amazon S3.
Launch mode (Mode de lancement)	Cluster	Cette option spécifie s'il faut lancer un cluster de longue durée ou un cluster qui s'interrompt après avoir exécuté toutes les étapes que vous spécifiez. Avec l'option Cluster, le cluster continue à s'exécuter jusqu'à ce que vous l'arrêtez. On appelle ceci un cluster de longue durée. Si vous choisissez Step execution (Exécution d'étape), Amazon EMR vous invite à ajouter et configurer des étapes. Vous pouvez utiliser des étapes pour soumettre du travail à un cluster. Le cluster s'arrête automatiquement une fois que les étapes que vous spécifiez terminent leur l'exécution. Pour plus d'informations, consultez Configuring a Cluster to Auto-Terminate or Continue (p. 110).

Champ de la console	Valeur par défaut	Description
Version	*emr-5.30.1*	Cette option spécifie la version Amazon EMR à utiliser lors de la création du cluster. La version Amazon EMR détermine la version des applications open source, telles que Hadoop et Hive, qu'Amazon EMR installe. L'étiquette de la dernière version est sélectionnée par défaut. Vous pouvez sélectionner une version d'Amazon EMR antérieure si vous avez besoin de différentes versions d'applications open source pour plus de compatibilité avec votre solution. Certaines fonctionnalités et applications Amazon EMR peuvent ne pas être disponibles lorsque vous utilisez des version antérieures d'Amazon EMR. Par conséquent et dans la mesure du possible, il est recommandé d'utiliser la dernière version. Pour plus d'informations sur chaque version Amazon EMR, consultez Amazon EMR Guide de version.

Champ de la console	Valeur par défaut	Description
Applications	Core Hadoop (Noyau Hadoop)	Cette option détermine les applications open source de l'écosystème Big Data à installer sur votre cluster. Les combinaisons d'applications les plus communes sont disponibles à l'aide du démarrage rapide. Pour sélectionner votre propre combinaison d'applications, y compris les applications supplémentaires ne figurant pas au démarrage rapide, choisissez Go to advanced options (Accéder aux options avancées). Pour plus d'informations sur les applications et les versions disponibles avec chaque version Amazon EMR, consultez la section Amazon EMR Guide de version. En outre, si une application n'est pas disponible pour être installée par Amazon EMR ou si vous devez installer une application personnalisée sur toutes les instances de cluster, vous pouvez utiliser une action d'amorçage. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125). Si vous sélectionnez Exécution d'étape, Amazon EMR choisit les applications à installer en fonction de ce que vos étapes requièrent.
Type d'instance	m5.xlarge	Cette option détermine le type d'instance Amazon EC2 qu'Amazon EMR initialise pour les instances qui s'exécutent dans votre cluster. La sélection d'instance par défaut varie d'une région à l'autre et certains types d'instances peuvent ne pas être disponibles dans certaines régions. Pour plus d'informations, consultez Configuration du matériel et de la mise en réseau d'un cluster (p. 129).

Champ de la console	Valeur par défaut	Description
Nombre d'instances	3	Cette option détermine le nombre d'instances Amazon EC2 à initialiser. Chaque instance correspond à un nœud dans le cluster Amazon EMR. Vous devez avoir au moins un nœud, le nœud principal. Pour obtenir des conseils sur le choix des types d'instance et le nombre d'instances, consultez Consignes et bonnes pratiques pour la configuration de cluster (p. 161).
Paire de clés EC2	Choose an option (Choisir une option)	Cette option spécifie la paire de clés Amazon EC2 à utiliser lorsque vous vous connectez aux nœuds de votre cluster à l'aide d'une connexion SSH (Secure Shell). Nous vous recommandons vivement de créer et de spécifier une paire de clés Amazon EC2. Si vous ne sélectionnez pas de paire de clés, vous ne pouvez pas vous connecter au cluster pour soumettre des étapes ou interagir avec les applications. Pour plus d'informations, consultez Connexion au cluster (p. 367). Pour vous connecter, vous devez également créer une règle entrante dans le groupe de sécurité pour autoriser les connexions SSH.
Autorisations	Par défaut	Utilisez cette option pour spécifier les rôles AWS Identity and Access Management utilisés par le cluster. Ces rôles déterminent les autorisations avec lesquelles Amazon EMR et les applications qui s'exécutent sur les instances de cluster doivent interagir avec d'autres services AWS. Vous pouvez choisir Custom (Personnalisé) pour spécifier vos propres rôles. Nous vous recommandons d'utiliser les rôles par défaut pour commencer. Pour plus d'informations, consultez Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).

Étape 3 : Autoriser les connexions SSH vers le cluster à partir de votre client

Les groupes de sécurité agissent en tant que pares-feux virtuels pour contrôler le trafic entrant et sortant de votre cluster. Lorsque vous créez votre premier cluster, Amazon EMR crée le groupe de sécurité par défaut géré par Amazon EMR associé à l'instance principale, ElasticMapReduce-master, et le groupe de sécurité associé aux nœuds de base et de tâche, ElasticMapReduce-slave.

> **Warning**
>
> La valeur par défaut du groupe de sécurité géré par EMR pour l'instance principale dans les sous-réseaux publics, `ElasticMapReduce-master`, est préconfigurée avec une règle qui autorise le trafic entrant sur le port 22 à partir de toutes les sources (IPv4 0.0.0.0/0). Cela permet de simplifier les connexions clients SSH initiales au nœud maître. Nous vous recommandons vivement de modifier cette règle entrante afin de restreindre le trafic uniquement aux sources fiables ou de spécifier un groupe de sécurité personnalisé qui limite l'accès.

La modification des groupes de sécurité n'est pas obligatoire pour terminer le didacticiel, mais nous vous recommandons de ne pas autoriser le trafic entrant à partir de toutes les sources. En outre, si un autre utilisateur a modifié le groupe de sécurité ElasticMapReduce-master pour éliminer cette règle conformément aux recommandations, vous n'êtes pas en mesure d'accéder au cluster à l'aide de SSH pour les prochaines étapes. Pour plus d'informations sur les groupes de sécurité, consultez Contrôle du trafic réseau avec des groupes de sécurité (p. 301) et Groupes de sécurité pour votre VPC dans le Amazon VPC Guide de l'utilisateur.

Pour supprimer la règle entrante qui autorise l'accès public via SSH pour le groupe de sécurité ElasticMapReduce-master

La procédure suivante suppose que le groupe de sécurité ElasticMapReduce-master n'a pas encore été modifié. De plus, pour modifier les groupes de sécurité, vous devez être connecté à AWS en tant qu'utilisateur racine ou principal IAM autorisé à gérer les groupes de sécurité pour le VPC dans lequel figure le cluster. Pour plus d'informations, consultez Modification des autorisations pour un utilisateur IAM et l'exemple de stratégie qui permet de gérer les groupes de sécurité EC2 dans le IAM Guide de l'utilisateur.

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Clusters.
3. Choisissez le Name (Nom) du cluster.
4. Sous Security and access (Sécurité et accès), choisissez le lien Security groups for Master (Groupes de sécurité pour le principal).

5. Choisissez ElasticMapReduce-master dans la liste.
6. Choisissez Entrant, Modifier.
7. Recherchez la règle dotée des paramètres suivants et choisissez l'icône x pour la supprimer :
   - Type

     SSH
   - Port

     22
   - Source

     Personnalisé 0.0.0.0/0
8. Faites défiler l'écran jusqu'au bas de la liste de règles et choisissez Add rule (Ajouter une règle).
9. Dans Type, sélectionnez SSH.

   Le champ Protocol (Protocole) est automatiquement défini sur TCP et Port Range (Plage de ports) sur 22.
10. Pour la source, sélectionnez My IP (Mon IP).

    Cette action ajoute automatiquement l'adresse IP de votre ordinateur client comme adresse source. Sinon, vous pouvez ajouter une plage d'adresses IP de clients approuvés Custom (Personnalisée) et choisir Add rule (Ajouter une règle) pour créer des règles supplémentaires pour d'autres clients. Dans de nombreux environnements de réseau, les adresses IP sont attribuées dynamiquement. Par conséquent, vous devrez peut-être modifier régulièrement les règles de groupe de sécurité pour mettre à jour l'adresse IP de confiance des clients.
11. Choisissez Save.
12. Choisissez éventuellement ElasticMapReduce-slave dans la liste et répétez les étapes ci-dessus pour autoriser le client SSH à accéder aux nœuds principaux et de tâches des clients de confiance.

Étape 4 : Traitez des données en exécutant le script Hive en tant qu'étape

Avec votre cluster en cours d'exécution, vous pouvez désormais soumettre un script Hive. Dans ce didacticiel, vous soumettez le script Hive en tant qu'étape à l'aide de la console Amazon EMR. Dans Amazon EMR, une étape est une unité de travail contenant une ou plusieurs tâches. Comme vous l'avez appris dans Étape 2 : Lancement de votre exemple de cluster Amazon EMR (p. 13), vous pouvez soumettre des étapes à un cluster de longue durée, ce que nous faisons dans cette étape. Vous pouvez également spécifier des étapes lorsque vous créez un cluster ou vous pouvez vous connecter au nœud principal, créer le script dans le système de fichiers local et l'exécuter à l'aide de la ligne de commande, par exemple `hive -f Hive_CloudFront.q`.

Présentation des données et du script

Les exemples de données et de script que vous utilisez dans ce didacticiel sont déjà disponibles dans un emplacement Amazon S3 auquel vous pouvez accéder.

Les exemples de données correspondent à une série de fichiers journaux d'accès Amazon CloudFront. Pour plus d'informations sur CloudFront et les formats de fichiers journaux , consultez Amazon CloudFront Manuel du développeur. Les données sont stockées dans Amazon S3 à l'adresse **`s3://`*`region`*`.elasticmapreduce.samples/cloudfront/data`** où *`région`* est votre région, par exemple, *`us-west-2`*. Lorsque vous saisissez l'emplacement au moment de soumettre l'étape, vous omettez le passage `cloudfront/data`, car le script l'ajoute.

Chaque entrée dans les fichiers journaux CloudFront fournit des détails sur une demande utilisateur unique au format suivant :

```
2014-07-05 20:00:00 LHR3 4260 10.0.0.15 GET eabcd12345678.cloudfront.net /test-
image-1.jpeg 200 - Mozilla/5.0%20(MacOS;%20U;%20Windows%20NT%205.1;%20en-US;
%20rv:1.9.0.9)%20Gecko/2009040821%20IE/3.0.9
```

L'exemple de script calcule le nombre total de demandes par système d'exploitation sur une période donnée. Le script utilise HiveQL, qui est un langage de script de type SQL pour l'entreposage et l'analyse des données. Le script est stocké dans Amazon S3 à l'adresse **`s3://`*`region`*`.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q`** où *`région`* est votre région.

L'exemple de script Hive effectue les opérations suivantes :

- Crée un schéma de table Hive nommé `cloudfront_logs`. Pour plus d'informations sur les tables Hive, consultez le didacticiel Hive sur le wiki Hive.
- Utilise le sérialiseur/désérialiseur d'expression régulière (RegEx SerDe) pour analyser les données d'entrée et appliquer le schéma de table. Pour plus d'informations, consultez SerDe sur le wiki Hive.
- Exécute une requête HiveQL contre la table `cloudfront_logs` et écrit les résultats de la requête à l'emplacement de sortie Amazon S3 que vous spécifiez.

Le contenu du script `Hive_CloudFront.q` est affiché ci-dessous. Les variables `${INPUT}` et `${OUTPUT}` sont remplacées par les emplacements Amazon S3 que vous spécifiez lorsque vous soumettez le script comme étape. Lorsque vous avez des données de référence dans Amazon S3, comme c'est le cas pour ce script, Amazon EMR utilise le système de fichiers EMR (EMRFS) pour lire les données d'entrée et écrire des données de sortie.

```
-- Summary: This sample shows you how to analyze CloudFront logs stored in S3 using Hive
```

```
-- Create table using sample data in S3.  Note: you can replace this S3 path with your own.
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (
  DateObject Date,
  LocalTime STRING,
  Location STRING,
  Bytes INT,
  RequestIP STRING,
  Method STRING,
  Host STRING,
  Uri STRING,
  Status INT,
  Referrer STRING,
  OS String,
  Browser String,
  BrowserVersion String
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "^(?!#)([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s
+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+([^ ]+)\\s+[^\(]+[\(]([^\;]+).*\%20([^\/]+)[\/](.*)$"
) LOCATION '${INPUT}/cloudfront/data';

-- Total requests per operating system for a given time frame
INSERT OVERWRITE DIRECTORY '${OUTPUT}/os_requests/' SELECT os, COUNT(*) count FROM
 cloudfront_logs WHERE dateobject BETWEEN '2014-07-05' AND '2014-08-05' GROUP BY os;
```

Soumission du script Hive en tant qu'étape

Utilisez l'option Add Step (Ajouter une étape) pour soumettre votre script Hive au cluster à l'aide de la console. Le script Hive et les exemples de données ont été chargés sur Amazon S3. Vous spécifiez l'emplacement de sortie en tant que dossier que vous avez précédemment créé dans Création d'un compartiment Amazon S3 (p. 13).

Pour exécuter le script Hive en le soumettant en tant qu'étape

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Dans Cluster List (Liste de clusters), sélectionnez le nom de votre cluster. Assurez-vous que le cluster se trouve dans un état d'Waiting (Attente).
3. Cliquez sur Steps (Étapes), puis sur Add step (Ajouter une étape).
4. Configurez l'étape en fonction des consignes suivantes :
   - Pour Step type (Type d'étape), choisissez Hive program (Programme Hive).
   - Pour Name (Nom), vous pouvez laisser la valeur par défaut ou saisir un nouveau nom. Si vous avez de nombreuses étapes dans un cluster, le nom vous aide à en garder une trace.
   - Pour Script S3 location (Emplacement de script S3), saisissez **`s3://`*`region`*`.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q`**. Remplacez *`région`* par votre identifiant de région. Par exemple, **`s3://us-west-2.elasticmapreduce.samples/cloudfront/code/Hive_CloudFront.q`** si vous travaillez dans la région Oregon. Pour obtenir la liste des régions et des identificateurs de région correspondants, consultez Régions et points de terminaison AWS pour Amazon EMR dans le AWS General Reference.
   - Pour Input S3 location (Emplacement d'entrée S3), saisissez **`s3://`*`region`*`.elasticmapreduce.samples`**.

     Remplacez *`région`* par votre identifiant de région.
   - Pour Output S3 location (Emplacement S3 de sortie), tapez ou recherchez le compartiment `output` que vous avez créé dans Création d'un compartiment Amazon S3 (p. 13).

- Pour Action on failure (Action sur échec), acceptez l'option par défaut Continue (Continuer). Cela spécifie que si l'étape échoue, le cluster continue à s'exécuter et traite les étapes suivantes. L'option Cancel and wait (Annuler et attendre) spécifie qu'un échec d'étape doit être annulé, que des étapes suivantes ne doivent pas être lancées, mais que le cluster doit continuer à fonctionner. L'option Terminate cluster (Arrêter le cluster) spécifie que le cluster doit s'arrêter si l'étape échoue.

5. Choisissez Add (Ajouter). L'étape s'affiche dans la console avec le statut Pending (En suspens).
6. Le statut de l'étape passe de Pending (En suspens) à Running (En cours d'exécution) puis à Completed (Terminé), au fur et à mesure de son exécution. Pour mettre à jour le statut, choisissez l'icône d'actualisation à droite du Filter (Filtre). Le script prend environ une minute pour s'exécuter.

Affichage des résultats

Une fois que l'étape a réussi, la sortie de requête Hive est enregistrée en tant que fichier texte dans le dossier de sortie Amazon S3 que vous avez spécifié lorsque vous avez soumis l'étape.

Pour afficher la sortie du script Hive

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez le Bucket name (Nom du compartiment) puis le dossier que vous avez créé précédemment. Par exemple, `moncompartiment` puis `MesRésultatsRequêteHive`.
3. La requête écrit les résultats dans un dossier à l'intérieur de votre dossier de sortie nommée os_requests. Choisissez ce dossier. Il devrait y avoir un fichier unique nommé 000000_0 dans le dossier. Il s'agit d'un fichier texte qui contient les résultats de votre requête Hive.
4. Choisissez le fichier, puis choisissez Download (Télécharger) pour enregistrer le fichier localement.
5. Utilisez l'outil de traitement de texte de votre choix pour ouvrir le fichier. Le fichier de sortie affiche le nombre de demandes d'accès organisées par système d'exploitation. L'exemple suivant illustre la sortie dans WordPad :

Étape 5 : Résilier le cluster et supprimer le compartiment

Une fois que vous avez terminé le didacticiel, vous pouvez mettre fin à votre cluster et supprimer votre compartiment Amazon S3 afin d'éviter des frais supplémentaires.

La mise hors service de votre cluster arrête les instances Amazon EC2 associées et interrompt l'accumulation des frais Amazon EMR. Amazon EMR conserve les informations de métadonnées sur des clusters terminés pour votre référence, sans frais, pendant deux mois. La console ne fournit aucun moyen de supprimer des clusters terminés afin qu'ils ne soient pas visibles dans la console. Les clusters résiliés sont supprimés du cluster lorsque les métadonnées sont supprimées.

Pour résilier le cluster

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Clusters, choisissez le cluster, puis choisissez Terminate (Résilier).

   Les clusters sont souvent créés avec la protection de la résiliation accidentelle activée, ce qui permet d'éviter la fermeture. Si vous avez bien suivi le didacticiel, la protection contre la résiliation doit être désactivée. Si la protection contre la résiliation est activée, vous êtes invité à en modifier le paramètre par précaution avant de résilier le cluster. Choisissez Change (Changer), Off (Désactivé).

Pour supprimer le compartiment de sortie

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez le compartiment dans la liste, de sorte que l'ensemble de ligne du compartiment soit sélectionné.
3. Choisissez de supprimer le compartiment, tapez le nom du compartiment, puis cliquez sur Confirm (Confirmer).

   Pour plus d'informations sur la suppression de dossiers et de compartiments, consultez Comment supprimer un compartiment S3 dans le Amazon Simple Storage Service Guide de mise en route.

Utilisation Amazon EMR Carnets

Vous pouvez utiliser Blocs-notes Amazon EMR avec Amazon EMR clusters en cours Apache Spark pour créer et ouvrir Jupyter Interface Notebook et jupyterlab dans le Amazon EMR console. Un Bloc-notes EMR est un bloc-notes « sans serveur » que vous pouvez utiliser pour exécuter des requêtes et du code. Contrairement à un carnet traditionnel, le contenu d'un Bloc-notes EMR elle-même—les équations, requêtes, modèles, code et texte narratif dans les cellules portables—exécuter dans un client. Les commandes sont exécutées à l'aide d'un noyau sur le cluster EMR. Le contenu des blocs-notes est également sauvegardé sur Amazon S3 séparément des données du cluster pour une durabilité et une réutilisation flexible.

Vous pouvez démarrer un cluster, joindre un Bloc-notes EMR pour l'analyse, puis mettre hors service le cluster. Vous pouvez également fermer un bloc-notes relié à un cluster en cours d'exécution et basculer vers un autre. Plusieurs utilisateurs peuvent joindre simultanément des ordinateurs portables au même cluster et partager des fichiers de bloc-notes dans Amazon S3 entre eux. Ces fonctionnalités vous permettent d'exécuter des clusters à la demande afin d'économiser des coûts et de réduire le temps consacré à la reconfiguration des blocs-notes pour différents clusters et ensembles de données.

Vous pouvez également exécuter un Bloc-notes EMR à l'aide de l'API EMR, sans avoir besoin d'interagir avec la console EMR (« exécution sans tête »). Vous devez inclure une cellule dans le Bloc-notes EMR qui possède un marqueur de paramètres. Cette cellule permet à un script de transmettre de nouvelles valeurs d'entrée au carnet. Les carnets paramétrés peuvent être réutilisés avec différents ensembles de valeurs d'entrée. Il n'est pas nécessaire de faire des copies du même carnet pour modifier et exécuter avec de nouvelles valeurs d'entrée. EMR crée et enregistre le carnet de sortie sur S3 pour chaque exécution du carnet paramétré. Pour Bloc-notes EMR Échantillons de code API, voir Commandes d'échantillons à exécuter Blocs-notes EMR programmatique (p. 33).

> **Important**
>
> Blocs-notes EMR est pris en charge avec les clusters créés à l'aide de la version Amazon EMR 5.18.0 et ultérieure. Nous vous recommandons vivement d'utiliser Blocs-notes EMR avec des clusters créés en utilisant la dernière version de Amazon EMR–particulièrement Amazon EMR version 5.30.0 et ultérieure, à l'exclusion de 6.0.0. Avec Amazon EMR 5.30.0, une modification a été apportée afin que les noyaux Jupyter s'exécutent sur le cluster attaché, plutôt que sur une instance Jupyter. Cette modification permet d'améliorer les performances et votre capacité à personnaliser les noyaux et les bibliothèques. Pour plus d'informations, consultez Différences dans les capacités selon la version de cluster (p. 26).

Frais applicables pour Amazon S3 stockage et pour Amazon EMR les clusters s'appliquent.

Considérations relatives à l'utilisation d'Blocs-notes EMR

Tenez compte des exigences suivantes lorsque vous créez des clusters et développez des solutions à l'aide de Bloc-notes EMR.

Exigences du cluster

- Activer Amazon EMR Bloquer l'accès public – L'accès entrant à un cluster permet aux utilisateurs du cluster d'exécuter des nels de carnet. Assurez-vous que seuls les utilisateurs autorisés peuvent accéder au cluster. Nous vous recommandons fortement de laisser l'accès public aux blocs activé et de limiter le trafic SSH entrant aux sources fiables uniquement. Pour plus d'informations, voir Utilisation d'Amazon EMR Block Public Access (p. 312) et Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

- Utiliser un cluster compatible – Un cluster relié à un cahier doit répondre aux exigences suivantes :
  - Seuls les clusters créés à l'aide de Amazon EMR sont pris en charge. Vous pouvez créer un cluster indépendamment Amazon EMR puis fixez Bloc-notes EMR, ou vous pouvez créer un cluster compatible lorsque vous créez un Bloc-notes EMR.
  - Seuls les clusters créés à l'aide de Amazon EMR version 5.18.0 et ultérieure sont pris en charge. Veuillez consulter the section called “Différences dans les capacités selon la version de cluster” (p. 26).
  - Clusters créés à l’aide d’Amazon EC2 avec processeurs AMD EPYC—par exemple, m5a.* et r5a.* types d’instances—ne sont pas pris en charge.
  - Blocs-notes EMR fonctionne uniquement avec les clusters créés avec `VisibleToAllUsers` réglé sur `true`. `VisibleToAllUsers` est `true` par défaut. Pour plus d'informations, consultez Présentation du paramètre EMR Cluster VisibleToAllUsers (p. 213).
  - Le cluster doit être lancé dans un EC2-VPC. Les sous-réseaux publics et privés sont pris en charge. La plateforme EC2-Classic n'est pas prise en charge.
  - Le cluster doit être lancé avec Hadoop, Spark et Livy installés. D'autres applications peuvent être installées, mais Blocs-notes EMR prend actuellement en charge uniquement les clusters Spark.
  - Les clusters avec authentification Kerberos ne sont pas pris en charge.
  - Les clusters intégrés à AWS Lake Formation prennent en charge l'installation de bibliothèques à portée de bloc-notes uniquement. L'installation des noyaux et des bibliothèques sur le cluster n'est pas prise en charge.
  - Les clusters avec plusieurs nœuds maîtres ne sont pas pris en charge.

Différences dans les capacités selon la version de cluster

Nous vous recommandons vivement d’utiliser Blocs-notes EMR avec des clusters créés en utilisant Amazon EMR version 5.30.0 ou ultérieure, à l’exclusion de 6.0.0. Avec ces groupes de versions, Blocs-notes EMR permet d’exécuter des fonels sur le joint Amazon EMR groupe. Les noyaux et les bibliothèques peuvent être installés directement sur le nœud maître du cluster. L'utilisation de Blocs-notes EMR avec ces versions de cluster présente les avantages suivants :

- Performance améliorée – Les nels de carnet fonctionnent sur des clusters avec des types d’instances EC2 que vous sélectionnez. Les versions antérieures exécutent des noyaux sur une instance spécialisée qui ne peut pas être redimensionnée, accessible ou personnalisée.
- Possibilité d’ajouter et de personnaliser des nels – Vous pouvez vous connecter au cluster pour installer des paquets kernel en utilisant `conda` et `pip`. En outre, `pip` l’installation est prise en charge par les commandes de terminal dans les cellules portables. Dans les versions antérieures, seuls les noyaux pré-installés étaient disponibles (Python, PySpark, Spark et SparkR). Pour plus d'informations, consultez Installation des noyaux et des bibliothèques Python sur un nœud maître de cluster (p. 43).
- Possibilité d’installer des bibliothèques Python – Vous pouvez installer les bibliothèques Python sur le nœud maître cluster (p. 43) utiliser `conda` et `pip`. Nous vous recommandons d’utiliser `conda`. Avec les versions antérieures, uniquement bibliothèques de carnet d’adresses (p. 43) pour pyspark sont pris en charge.

Fonctionnalités Blocs-notes EMR prises en charge par la version du cluster

Version de cluster	Bibliothèques à portée de bloc-notes pour PySpark	Installation du noyau sur le cluster	Installation de la bibliothèque Python sur le nœud maître
Antérieur à 5.18.0	Blocs-notes EMR non pris en charge		

Version de cluster	Bibliothèques à portée de bloc-notes pour PySpark	Installation du noyau sur le cluster	Installation de la bibliothèque Python sur le nœud maître
5.18.0–5.25.0	Non.	Non.	Non.
5.26.0–5.29.0	: oui (p. 43)	Non.	Non.
5.30.0	: oui (p. 43)	: oui (p. 43)	: oui (p. 43)
6.0.0	Non.	Non.	Non.

Limites pour les ordinateurs portables connectés simultanément

Prenez en compte le type d'instance EC2 du nœud principal du cluster lorsque vous créez un cluster qui prend en charge les blocs-notes. Les contraintes de mémoire de cette instance EC2 déterminent le nombre de blocs-notes qui peuvent être prêts simultanément pour exécuter du code et des demandes sur le cluster.

Type d'instance EC2 de nœud principal	Nombre de blocs-notes
medium	2
large	4
xlarge	8
2xlarge	16
*.4xlarge	24
*.8xlarge	24
*.16xlarge	24

Versions bloc-notes Jupyter et Python

Blocs-notes EMR exécuter Carnet Jupyter version 6.0.2 et Python 3.6.5, quel que soit le Amazon EMR version de la version du cluster ci-joint.

Si vous spécifiez un emplacement crypté dans Amazon S3 pour stocker les fichiers du carnet, vous devez configurer le Service Role for Blocs-notes EMR (p. 225) en tant qu'utilisateur clé. Le rôle de service par défaut est `EMR_Notebooks_DefaultRole`. Si vous utilisez un AWS KMS clé pour chiffrement, voir Utilisation des politiques clés dans AWS KMS dans le AWS Key Management Service Developer Guide et le article d'assistance pour l'ajout d'utilisateurs clés.

Création d'un bloc-notes

Vous créez un Bloc-notes EMR utiliser le Amazon EMR console. Création de carnets à l'aide du AWS CLI ou le Amazon EMR API n'est pas pris en charge.

Pour créer un Bloc-notes EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisir Carnets, Créer un carnet.
3. Saisissez un Nom du carnet et une option Description du carnet.
4. Si vous disposez d'un cluster actif exécutant Hadoop, Spark et Livy pour lequel vous souhaitez joindre le bloc-notes, laissez la valeur par défaut Choisir un cluster existant sélectionné, cliquez sur Choisir, sélectionnez un cluster dans la liste, puis cliquez sur Choisir le cluster. Seuls les clusters qui répondent aux exigences apparaissent. Pour plus d'informations, consultez Considérations relatives à l'utilisation d'Blocs-notes EMR (p. 25).

   —ou—

   Choisir Créer un cluster, entrez un Nom du cluster et choisissez les options conformément aux directives suivantes. Le cluster est créé dans le VPC par défaut pour le compte à l'aide d'instances à la demande.

Paramètre	Description :
Nom du cluster	Nom convivial utilisé pour identifier le cluster.
Version : ()	Impossible de modifier. Par défaut, la version Amazon EMR la plus récente (5.30.1).
: Applications	Impossible de modifier. Répertorie les applications installées sur le cluster.
Instance	Entrez le nombre d'instances et sélectionnez le type d'instance EC2. Une instance est utilisée pour le nœud maître. Les autres sont utilisées pour les nœuds principaux. Le type d'instance détermine le nombre de blocs-notes pouvant être attachés au cluster simultanément. Pour plus d'informations, consultez Limites pour les ordinateurs portables connectés simultanément (p. 27).
Rôle EMR	Laissez la valeur par défaut ou choisissez le lien pour lequel spécifier un rôle de service personnalisé Amazon EMR. Pour plus d'informations, consultez Service Role for Amazon EMR (EMR Role) (p. 218).
Profil d'instance EC2	Laissez la valeur par défaut ou choisissez le lien pour spécifier un rôle de service personnalisé pour les instances EC2. Pour plus d'informations, consultez Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).
Paire de clés EC2	Choisissez une paire de clés EC2 pour pouvoir vous connecter à des instances de cluster. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368).

5. Pour Groupes de sécurité, choisissez Utiliser les groupes de sécurité par défaut. Sinon, choisissez Choisir des groupes de sécurité et sélectionnez des groupes de sécurité personnalisés disponibles dans le VPC du cluster. Vous en sélectionnez un pour l'instance principale et un autre pour l'instance

client de bloc-notes. Pour plus d'informations, consultez the section called “Groupes de sécurité pour Blocs-notes EMR” (p. 311).

6. Pour Rôle de service AWS, laissez la valeur par défaut ou choisissez un rôle personnalisé dans la liste. L'instance client du bloc-notes utilise ce rôle. Pour plus d'informations, consultez Service Role for Blocs-notes EMR (p. 225).
7. Pour Emplacement du carnet choisir l’emplacement dans Amazon S3 où le fichier du carnet est enregistré ou spécifier votre propre emplacement. Si le compartiment et le dossier n'existent pas, Amazon EMR le crée.

   Amazon EMR crée un dossier avec ID notebook comme nom de dossier, et enregistre le carnet dans un fichier nommé `NotebookName.ipynb`. Par exemple, si vous spécifiez Amazon S3 emplacement `s3://MyBucket/MyNotebooks` pour un ordinateur portable nommé `MyFirstEMRManagedNotebook`, le fichier du carnet est enregistré sur `s3://MyBucket/MyNotebooks/NotebookID/MyFirstEMRManagedNotebook.ipynb`.

   Si vous spécifiez un emplacement crypté dans Amazon S3, vous devez configurer le Service Role for Blocs-notes EMR (p. 225) en tant qu’utilisateur clé. Le rôle de service par défaut est `EMR_Notebooks_DefaultRole`. Si vous utilisez un AWS KMS clé pour chiffrement, voir Utilisation des politiques clés dans AWS KMS dans le AWS Key Management Service Developer Guide et le article d’assistance pour l’ajout d’utilisateurs clés.
8. Éventuellement, si vous avez ajouté un référentiel Git à Amazon EMR que vous souhaitez associer à ce carnet, choisissez Référentiel Git, cliquez Choisir un référentiel puis sélectionnez un référentiel dans la liste. Pour plus d'informations, consultez Association de référentiels Git à Blocs-notes EMR (p. 44).
9. Facultatif, choisissez Tags, puis ajoutez toutes les étiquettes de valeur clé supplémentaires pour le carnet.

   Important

   Une balise par défaut avec Clé chaîne définie sur `creatorUserID` et la valeur définie sur votre ID utilisateur IAM est appliquée à des fins d’accès. Nous vous recommandons de ne pas modifier ou supprimer cette balise, car elle peut être utilisée pour contrôler l'accès. Pour plus d'informations, consultez Utiliser les balises de cluster et de bloc-notes avec des stratégies de contrôle d'accès IAM (p. 212).
10. Choisir Créer un carnet.

Utilisation des blocs-notes

Une fois que vous avez créé une Bloc-notes EMR, le bloc-notes prend un peu de temps pour démarrer. Le Statut dans le Carnets afficher la liste Démarrage. Vous pouvez ouvrir un ordinateur portable lorsque son statut est Prêt. Il peut s’avérer un peu plus long pour qu’un ordinateur portable soit Prêt si vous avez créé un cluster avec.

Tip

Actualisez votre navigateur ou choisissez l'icône d'actualisation au-dessus de la liste des blocs-notes pour actualiser le statut du bloc-notes.

Comprendre le statut de bloc-notes

Un Bloc-notes EMR peuvent avoir les éléments suivants : Statut dans le Carnets.

Statut	Signification
Prêt	Vous pouvez ouvrir le bloc-notes à l'aide de l'éditeur de bloc-notes. Alors qu’un ordinateur

Statut	Signification
	portable a un Prêt vous pouvez l'arrêter ou la supprimer. Vous devez d'abord arrêter le bloc-notes pour modifier les clusters. Si un carnet de notes dans le Prêt Le statut est inactif pendant une longue période, il est arrêté automatiquement.
Démarrage en cours	Le bloc-notes est en cours d'être créé et joint au cluster. Lorsqu'un bloc-notes est en cours de démarrage, vous ne pouvez pas supprimer l'éditeur de bloc-notes, l'arrêter, le supprimer ou modifier les clusters.
En suspens	Le bloc-notes a été créé et est en attente d'intégration avec le cluster pour terminer. Il se peut que le cluster soit toujours en train de mettre en service des ressources ou de répondre à d'autres demandes. Vous pouvez ouvrir l'éditeur de carnet avec le carnet dans mode local. Tout code qui s'appuie sur les processus de cluster n'est pas exécuté et échoue.
En cours d'arrêt	Le bloc-notes est en cours d'arrêt ou le cluster auquel le bloc-notes est attaché est en cours d'arrêt. Lorsqu'un bloc-notes est en cours d'arrêt, vous ne pouvez pas supprimer l'éditeur de bloc-notes, l'arrêter, le supprimer ou modifier les clusters.
Arrêté(e)	Le bloc-notes s'est arrêté. Vous pouvez démarrer le bloc-notes sur le même cluster, tant que le cluster est encore en cours d'exécution. Vous pouvez changer de clusters et supprimer le cluster.
Suppression en cours	Le cluster est en cours de suppression de la liste des clusters disponibles. Le fichier du carnet, `NotebookName.ipynb` reste dans Amazon S3 et continue de cumuler les frais de stockage applicables.

Utilisation de l'éditeur de bloc-notes

Un avantage de l'utilisation d'un Bloc-notes EMR est que vous pouvez lancer le bloc-notes dans Jupyter ou JupyterLab directement à partir de la console.

Avec Blocs-notes EMR, l'éditeur de carnet auquel vous accédez Amazon EMR console est l'éditeur de carnet Jupyter open source ou jupyterlab. Parce que l'éditeur de carnet est lancé dans le Amazon EMR console, il est plus efficace de configurer l'accès qu'avec un ordinateur portable hébergé sur un Amazon EMR groupe. Vous n'avez pas besoin de configurer un client de l'utilisateur pour l'accès web via SSH, les règles du groupe de sécurité et les configurations de proxy. Si un utilisateur dispose d'autorisations suffisantes, il peut simplement ouvrir l'éditeur de carnet dans le Amazon EMR console.

Seul un utilisateur à la fois peut avoir un Bloc-notes EMR ouvert dans Amazon EMR. Une erreur se produit si un autre utilisateur essaie d'ouvrir un Bloc-notes EMR qui est déjà ouvert.

Important

Amazon EMR crée une URL pré-signée unique pour chaque session d'édition bloc-notes, valable uniquement pendant un court instant. Nous vous recommandons de ne pas partager l'URL d'éditeur de bloc-notes. Cela créerai un risque de sécurité, car les destinataires de l'URL adoptent vos autorisations à modifier le bloc-notes et à exécuter le code de bloc-notes pour toute la durée de vie de l'URL. Si d'autres personnes ont besoin d'accéder à un cahier, donnez des autorisations à leur IAM l'utilisateur par le biais des politiques d'autorisations et s'assurer que le rôle de service pour Blocs-notes EMR a accès au Amazon S3. Pour plus d'informations, voir the section called "Sécurité" (p. 42) et Service Role for Blocs-notes EMR (p. 225).

Ouvrir l'éditeur de bloc-notes pour un Bloc-notes EMR

1. Sélectionnez un carnet avec un Statut de Prêt ou En attente du Carnets.
2. Choisir Ouvert à jupyterlab ou Ouvert à Jupyter.

   Un nouvel onglet de navigateur s'ouvre pour l'éditeur de bloc-notes JupyterLab ou Jupyter.
3. Du Noyau menu, choisissez Modifier le noyau puis sélectionnez le noyau de votre langue de programmation.

   Vous êtes maintenant prêt à écrire et exécuter du code à partir de l'éditeur de bloc-notes.

Enregistrement du contenu d'un bloc-notes

Lorsque vous travaillez dans l'éditeur de bloc-notes, le contenu des cellules et sorties de blocs-notes sont enregistrés automatiquement et périodiquement vers le fichier de bloc-notes Amazon S3. Un carnet qui n'a pas de changement depuis la dernière fois qu'une cellule a été modifiée (auto-enregistré) à côté du nom du carnet dans l'éditeur. Si les modifications n'ont pas encore été enregistrées, modifications non enregistrées apparaît.

Vous pouvez manuellement enregistrer un bloc-notes. Du Fichier menu, choisissez Enregistrer et point de contrôle ou appuyez sur CTRL+S. Cela crée un fichier nommé `NotebookName.ipynb` dans un points de contrôle dossier dans le dossier du carnet dans Amazon S3. Par exemple, `s3://MyBucket/MyNotebookFolder/NotebookID/checkpoints/NotebookName.ipynb`. Seul le fichier de point de contrôle le plus récent est enregistré dans cet emplacement.

Changement de clusters

Vous pouvez changer le cluster auquel un Bloc-notes EMR est attaché sans modifier le contenu du bloc-notes lui-même. Vous pouvez modifier les clusters uniquement pour les ordinateurs portables ayant un Arrêté statut.

Pour modifier le cluster d'un Bloc-notes EMR

1. Si le bloc-notes que vous souhaitez modifier est en cours d'exécution, sélectionnez-le à partir du Carnets et choisissez Arrêter.
2. Lorsque le statut du carnet est Arrêté, sélectionnez le carnet de notes du Carnets puis choisissez Afficher les détails.
3. Choisir Changer le cluster.
4. Si vous disposez d'un cluster actif exécutant Hadoop, Spark et Livy auquel vous souhaitez associer le bloc-notes, conservez la valeur par défaut et sélectionnez un cluster dans la liste. Seuls les clusters qui répondent aux exigences sont répertoriés.

   —ou—

Choisir Créer un cluster puis choisissez les options de cluster. Pour plus d'informations, consultez Exigences du cluster (p. 25).

5. Choisissez une option pour Groupes de sécurité, puis choisissez Changer le cluster et démarrer le carnet.

Suppression des blocs-notes et des fichiers de bloc-notes

Lorsque vous supprimez un Bloc-notes EMR utiliser le Amazon EMR , vous supprimez le carnet de notes de la liste des carnets disponibles. Cependant, les fichiers de bloc-notes restent dans Amazon S3 et continuent d'accumuler des charges de stockage.

Pour supprimer un bloc-notes et retirer les fichiers associés

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisir Carnets, sélectionnez votre carnet de notes dans la liste, puis choisissez Afficher les détails.
3. Cliquez sur l'icône du dossier à côté de Emplacement du carnet et copiez URL, qui se trouve dans le modèle `s3://MyNotebookLocationPath/NotebookID/`.
4. Choisir Supprimer.

   Le bloc-notes est supprimé de la liste et les détails de bloc-notes ne peuvent plus être consultés.
5. Suivez les instructions pour Comment supprimer les dossiers d'un bucket S3 dans le Amazon Simple Storage Service Guide de l'utilisateur de la console. Naviguez vers le compartiment et le dossier de l'étape 3.

   —ou—

   Si vous avez installé l'AWS CLI, ouvrez une invite de commande et tapez la commande à la fin de ce paragraphe. Remplacez l'emplacement Amazon S3 par l'emplacement que vous avez copié ci-dessus. Assurez-vous que AWS CLI est configuré avec les touches d'accès d'un utilisateur disposant des autorisations pour supprimer Amazon S3. Pour plus d'informations, voir Configuration du AWS CLI dans le AWS Command Line Interface Guide de l'utilisateur.

```
aws s3 rm s3://MyNotebookLocationPath/NotebookID
```

Partage de fichiers de bloc-notes

Chaque Bloc-notes EMR est enregistré sur Amazon S3 comme fichier nommé `NotebookName.ipynb`. Tant que le fichier notebook est compatible avec la même version de Jupyter Notebook que Blocs-notes EMR est basé sur le bouton, vous pouvez ouvrir le carnet comme Bloc-notes EMR.

La façon la plus simple d'ouvrir un fichier bloc-notes à partir d'un autre utilisateur est d'enregistrer le fichier*.ipynb d'un autre utilisateur sur votre système de fichiers local, puis d'utiliser la fonction de téléchargement dans les éditeurs Jupyter et JupyterLab.

Vous pouvez utiliser cette procédure pour utiliser les blocs-notes EMR partagés par d'autres, des blocs-notes Jupyter partagés dans la communauté, ou pour restaurer un bloc-notes qui a été supprimé de la console lorsque vous disposez encore du fichier bloc-notes.

Pour utiliser un autre fichier de bloc-notes en tant que base pour un Bloc-notes EMR

1. Avant de poursuivre, fermez l'éditeur de bloc-notes pour tous les blocs-notes que vous utilisez, puis arrêtez le bloc-notes s'il s'agit d'un Bloc-notes EMR.
2. Créez un Bloc-notes EMR et donnez-lui un nom. Le nom que vous saisissez pour le bloc-notes sera le nom du fichier que vous devez remplacer. Le nouveau nom de fichier doit correspondre exactement au nom de ce fichier.
3. Prenez note de l'emplacement Amazon S3 que vous choisissez pour le bloc-notes. Le fichier que vous remplacez est dans un dossier avec un chemin d'accès et un nom de fichier comme le modèle suivant : . `s3://MyNotebookLocation/NotebookID/MyNotebookName.ipynb`.
4. Arrêtez le bloc-notes.
5. Remplacez l'ancien fichier de bloc-notes dans l'emplacement Amazon S3 par le nouveau, en utilisant exactement le même nom.

   Les éléments suivants AWS CLI commande pour Amazon S3 remplace un fichier enregistré sur une machine locale appelée `SharedNotebook.ipynb` pour un Bloc-notes EMR avec le nom mynotebook, un ID de `e-12A3BCDEFJHIJKLMNO45PQRST`, et créé avec `MyBucket/MyNotebooksFolder` spécifié dans Amazon S3. Pour plus d'informations sur l'utilisation de Amazon S3 pour copier et remplacer les fichiers, voir Téléchargement, téléchargement et gestion des objets dans le Amazon Simple Storage Service Guide de l'utilisateur de la console.

```
aws s3 cp SharedNotebook.ipynb s3://MyBucket/
MyNotebooksFolder/-12A3BCDEFJHIJKLMNO45PQRST/MyNotebook.ipynb
```

Commandes d'échantillons à exécuter Blocs-notes EMR programmatique

Les API d'exécution du carnet DME sont disponibles pour exécuter les cahiers DME via un script ou une ligne de commande. La possibilité de démarrer, d'arrêter, de répertorier et de décrire les exécutions des ordinateurs portables EMR sans la console AWS vous permet de contrôler de manière programmatique un carnet d'EMR. À l'aide d'une cellule de carnet paramétrée, vous pouvez transmettre différentes valeurs de paramètres à un carnet sans avoir à créer une copie du carnet pour chaque nouvel ensemble de valeurs de paramètres. Voir Actions API EMR.

L'exécution du carnet DME peut être planifiée ou mise en lots à l'aide des événements AWS cloudwatch et AWS Lambda. Voir Utilisation d'AWS Lambda avec Amazon Cloudwatch Events

Cette section présente plusieurs exemples d'exécution programmatique du cahier DME à l'aide de l'interface de ligne de commande AWS, Boto3 SDK (Python) et Ruby.

Échantillons de commande de la CLI d'exécution du carnet (p. 33)

Exécution d'un notebook Échantillons Python (p. 36)

Échantillons de Rubis pour l'exécution du carnet (p. 38)

Échantillons de commande de la CLI d'exécution du carnet

Pour plus d'informations sur les actions d'exécution de l'API EMR, voir Actions API EMR.

```
aws emr --region us-east-1 \
start-notebook-execution \
--editor-id e-BKTM2DIHXBEDRU44ANWRKIU8N \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"]}' \
--relative-path test.ipynb \
--notebook-execution-name my-execution \
--execution-engine '{"Id" : "j-2QMOV6JAX1TS2"}' \
--service-role EMR_Notebooks_DefaultRole


{
    "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE"
}
```

Sortie de la commande CLI Bloc-notes EMR

Voici un exemple de carnet de sortie. Cellule [3] affiche les valeurs de paramètre nouvellement injectées.

```
In [1]:
print("Hello world")

Hello world

In [2]: parameters
input_param = "default"
good_superhero = ["batman", "superman"]

In [3]: injected-parameters
# Parameters
good_superhero = ["superman", "batman"]
input_param = "my-value"
new_param = {"nest-key1": "nest-val1", "nest-key2": "nest-val2"}

In [4]:
print(input_param)

my-value

In [5]:
for hero in good_superhero:
    print(hero)

superman
batman
```

Description de la commande CLI d'exécution du carnet

```
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR",
            "MasterInstanceSecurityGroupId": "sg-05ce12e58cd4f715e"
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":[\"superman\",
 \"batman\"]}",
        "Status": "FINISHED",
        "StartTime": 1593490857.009,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
```

```
        "LastStateChangeReason": "Execution is finished for cluster j-2QMOV6JAX1TS2.",
        "NotebookInstanceSecurityGroupId": "sg-0683b0a39966d4a6a",
        "Tags": []
    }
}
```

Arrêt de la commande CLI d'exécution du carnet

```
# stop a running execution
aws emr --region us-east-1 \
stop-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T


# describe it
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR"
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":[\"superman\",
 \"batman\"]}",
        "Status": "STOPPED",
        "StartTime": 1593490876.241,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:editor-execution/ex-
IZWZX78UVPAATC8LHJR129B1RBN4T",
        "LastStateChangeReason": "Execution is stopped for cluster j-2QMOV6JAX1TS2.
 Internal error",
        "Tags": []
    }
}
```

Mise en liste de l'exécution du carnet par commande CLI de démarrage

```
# filter by start time
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000

{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490876.241
        },
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "RUNNING",
            "StartTime": 1593490857.009
        },
```

```
        {
            "NotebookExecutionId": "ex-IZWZYRS0M14L5V95WZ9OQ399SKMNW",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490292.995
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593489834.765
        },
        {
            "NotebookExecutionId": "ex-IZWZXOZF88JWDF9J09GJ91R57VI0N",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FAILED",
            "StartTime": 1593488934.688
        }
    ]
}
```

Mise en liste de l'exécution du carnet par heure de début et commande CLI

```
# filter by start time and status
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000 --status FINISHED
{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593490857.009
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593489834.765
        }
    ]
}
```

Exécution d'un notebook Échantillons Python

L'exemple de code suivant est un fichier Boto3 (SDK Python AWS) appelé `demo.py` qui montre les API d'exécution du carnet.

Pour plus d'informations sur les actions d'exécution de l'API EMR, voir Actions API EMR.

```
    demo.py:

    import boto3,time
```

```
    emr = boto3.client(
        'emr',
        region_name='us-west-1'
    )



    start_resp = emr.start_notebook_execution(
     EditorId='e-40AC8ZO6EGGCPJ4DLO48KGGGI',
     RelativePath='boto3_demo.ipynb',
     ExecutionEngine={'Id':'j-1HYZS6JQKV11Q'},
     ServiceRole='EMR_Notebooks_DefaultRole'
    )

    execution_id = start_resp["NotebookExecutionId"]
    print(execution_id)
    print("\n")


    describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)

    print(describe_response)
    print("\n")



    list_response = emr.list_notebook_executions()
    print("Existing notebook executions:\n")
    for execution in list_response['NotebookExecutions']:
     print(execution)
     print("\n")



    print("Sleeping for 5 sec...")
    time.sleep(5)

    print("Stop execution " + execution_id)
    emr.stop_notebook_execution(NotebookExecutionId=execution_id)
    describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)
    print(describe_response)
    print("\n")
```

Voici la sortie de running demo.py.

```
    [ec2-user@ip-10-1-58-123 ~]$ python demo.py
    ex-IZX56YJDW1D29Q1PHR32WABU2SAPK


    {'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id': 'j-1HYZS6JQKV11Q',
 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STARTING', 'StartTime':
 datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()), 'Arn':
 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is starting for cluster
 j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId': '70f12c5f-1dda-45b7-
adf6-964987d373b7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid':
 '70f12c5f-1dda-45b7-adf6-964987d373b7', 'content-type': 'application/x-amz-json-1.1',
 'content-length': '448', 'date': 'Wed, 19 Aug 2020 00:49:22 GMT'}, 'RetryAttempts': 0}}


    Existing notebook executions:
```

```
    {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'STARTING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal())}


    {'NotebookExecutionId': 'ex-IZX5ABS5PR1E5AHMFYEMX3JJIORRB', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'RUNNING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 48, 36, 373000, tzinfo=tzlocal())}


    {'NotebookExecutionId': 'ex-IZX5GLVXIU1HNI8BWVW057F6MF4VE', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 45, 14, 646000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 46, 26, 543000, tzinfo=tzlocal())}


    {'NotebookExecutionId': 'ex-IZX5CV8YDUO8JAIWMXN2VH32RUIT1', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 43, 5, 807000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 44, 31, 632000, tzinfo=tzlocal())}


    {'NotebookExecutionId': 'ex-IZX5AS0PPW55CEDEURZ9NSOWSUJZ6', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 42, 29, 265000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 43, 48, 320000, tzinfo=tzlocal())}


    {'NotebookExecutionId': 'ex-IZX57YF5Q53BKWLR4I5QZ14HJ7DRS', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 38, 37, 81000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 40, 39, 646000, tzinfo=tzlocal())}


    Sleeping for 5 sec...
    Stop execution ex-IZX56YJDW1D29Q1PHR32WABU2SAPK
    {'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id': 'j-1HYZS6JQKV11Q',
 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STOPPING', 'StartTime':
 datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()), 'Arn':
 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is being stopped for
 cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId': '2a77ef73-
c1c6-467c-a1d1-7204ab2f6a53', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid':
 '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'content-type': 'application/x-amz-json-1.1',
 'content-length': '453', 'date': 'Wed, 19 Aug 2020 00:49:30 GMT'}, 'RetryAttempts': 0}}
```

Échantillons de Rubis pour l'exécution du carnet

Les exemples suivants sont des échantillons de code Rubis qui démontrent l'utilisation de l'API d'exécution du cahier.

```
# prepare an EMR client

emr = Aws::EMR::Client.new(
  region: 'us-east-1',
  access_key_id: 'AKIA...JKPKA',
  secret_access_key: 'rLMeu...vU0OLrAC1',
)
```

Démarrage de l'exécution du carnet et obtention de l'ID d'exécution

Dans cet exemple, l'éditeur S3 et le carnet DME sont `s3://mybucket/notebooks/e-EA8VGAA429FEQTC8HC9ZHWISK/test.ipynb`.

Pour plus d'informations sur les actions d'exécution de l'API EMR, voir Actions API EMR.

```
start_response = emr.start_notebook_execution({
    editor_id: "e-EA8VGAA429FEQTC8HC9ZHWISK",
    relative_path: "test.ipynb",

    execution_engine: {id: "j-3U82I95AMALGE"},

    service_role: "EMR_Notebooks_DefaultRole",
})


notebook_execution_id = start_resp.notebook_execution_id
```

Description de l'exécution du carnet et impression des détails

```
describe_resp = emr.describe_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
puts describe_resp.notebook_execution

{
:notebook_execution_id=>"ex-IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:editor_id=>"e-EA8VGAA429FEQTC8HC9ZHWISK",
:execution_engine=>{:id=>"j-3U82I95AMALGE", :type=>"EMR", :master_instance_security_group_id=>nil},
:notebook_execution_name=>"",
:notebook_params=>nil,
:status=>"STARTING",
:start_time=>2020-07-23 15:07:07 -0700,
:end_time=>nil,
:arn=>"arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:output_notebook_uri=>nil,
:last_state_change_reason=>"Execution is starting for cluster
 j-3U82I95AMALGE.", :notebook_instance_security_group_id=>nil,
:tags=>[]
}
```

Liste des exécutions de carnet

```
puts 'Listing executions:'
list_resp = emr.list_notebook_executions({})
puts list_resp
```

Filtres pour ordinateur portable

```
"EditorId": "e-XXXX",           [Optional]
"From" : "1593400000.000",    [Optional]
```

```
"To" : "1598400000.500",      [Optional]
"Status":                       [Optional]
        "START_PENDING | STARTING | RUNNING | STOP_PENDING | STOPPING | FINISHING | FAILING
 | FINSHED | FAILED | STOPPED"
```

Un carnet d'EMR peut être arrêté lorsqu'il est dans l'un de ces états : `START_PENDING, STARTING, RUNNING`.

Arrêt de l'exécution du carnet

```
stop_resp = emr.stop_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
```

Programmation des exécutions des ordinateurs portables

Les carnets DME peuvent être programmés en utilisant :

- Un travail de cron avec le script bash
- Un événement AWS Cloudwatch, qui déclenchera un AWS Lambda (Java/Python/Javascript). Pour plus d'informations, voir Utilisation d'AWS Lambda avec Amazon Cloudwatch Events.

```
#!/bin/bash
aws emr --region us-east-1 \
start-notebook-execution \
--editor-id e-EA8VGAA429FEQTC8HC9ZHWISK \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"],
 "new_param":{"nest-key1":"nest-val1", "nest-key2":"nest-val2"}}' \
--relative-path test.ipynb \
--notebook-execution-name daily-job \
--execution-engine '{"Id" : "j-3U82I95AMALGE"}' \
--service-role EMR_Notebooks_DefaultRole
```

Activation de l'emprunt d'identité utilisateur pour surveiller l'activité des utilisateurs et des tâches Spark

Blocs-notes EMR vous permet de configurer l'utilisation d'une identité d'un autre utilisateur sur un cluster Spark. Cette fonctionnalité vous permet de suivre les tâches d'activité lancées à partir de l'éditeur de bloc-notes. En outre, Blocs-notes EMR intègre un widget de bloc-notes Jupyter pour afficher des détails de tâche Spark avec la sortie de requête dans l'éditeur de bloc-notes. Le widget est disponible par défaut et ne nécessite aucune configuration spéciale. Toutefois, pour afficher les serveurs d'historique, votre client doit être configuré pour afficher les interfaces web Amazon EMR hébergées sur le nœud principal.

Configuration de l'emprunt d'identité d'un utilisateur Spark

Par défaut, les tâches Spark que les utilisateurs soumettent à l'aide de l'éditeur de bloc-notes semblent provenir d'une identité d'utilisateur `livy` indistincte. Vous pouvez configurer l'emprunt d'identité de

l'utilisateur pour le cluster afin que ces tâches soient associées à l'identité de l'utilisateur IAM qui a exécuté le code à la place. Les répertoires d'utilisateurs HDFS sur le nœud principal sont créés pour chaque identité d'utilisateur qui exécute du code dans le bloc-notes. Par exemple, si l'utilisateur `NbUser1` exécute le code à partir de l'éditeur de carnet, vous pouvez vous connecter au noeud principal et voir `hadoop fs -ls /user` affiche le répertoire `/user/user_NbUser1`.

Vous activez cette fonction en définissant des propriétés dans `core-site` et `livy-conf` classifications de configuration. Cette fonctionnalité n'est pas disponible par défaut lorsque vous demandez à Amazon EMR de créer un cluster avec un bloc-notes. Pour plus d'informations sur l'utilisation des classifications de configuration pour personnaliser les applications, voir Configuration des applications dans le Amazon EMR Guide de version.

Utilisez les classifications de configuration et les valeurs suivantes pour activer l'emprunt d'identité de l'utilisateur pour Blocs-notes EMR:

```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
        }
    }
]
```

Utilisation du widget de surveillance de tâche Spark

Lorsque vous exécutez du code dans l'éditeur de bloc-notes qui exécute les tâches Spark sur le cluster EMR, la sortie inclut un widget Jupyter Notebook pour la surveillance de tâche Spark. Le widget fournit des détails de la tâche et des liens utiles vers la page de serveur d'historique Spark et la page de l'historique des tâches Hadoop, ainsi que des liens pratiques vers les journaux de tâche dans Amazon S3 pour les tâches échouées.

Pour afficher les pages de serveur d'historique sur le nœud principal du cluster, vous devez configurer un client SSH et un proxy, le cas échéant. Pour plus d'informations, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372). Pour afficher les journaux dans Amazon S3, la journalisation de cluster doit être activée (la valeur par défaut pour les nouveaux clusters). Pour plus d'informations, consultez View Log Files Archived to Amazon S3 (p. 335).

Voici un exemple de surveillance de l'emploi Spark widget.

Sécurité et contrôle d'accès des blocs-notes EMR

Plusieurs fonctionnalités sont disponibles pour vous aider à adapter la sécurité de Blocs-notes EMR. Cela permet d'assurer que seuls les utilisateurs autorisés ont accès à un Bloc-notes EMR, peuvent travailler avec des blocs-notes et peuvent utiliser l'éditeur de bloc-notes pour exécuter du code sur le cluster. Ces fonctions fonctionnent avec les fonctions de sécurité disponibles pour Amazon EMR et Amazon EMR clusters. Pour plus d'informations, consultez Sécurité dans Amazon EMR (p. 177).

- Vous pouvez utiliser des déclarations de stratégie AWS Identity and Access Management avec les balises de bloc-notes pour limiter l'accès. Pour plus d'informations, voir Clés de condition (p. 212) et Exemple de déclarations de stratégie basées sur les identités pour Blocs-notes EMR (p. 248).
- Les groupes de sécurité EC2 agissent comme un pare-feu virtuel qui contrôle le trafic réseau entre l'instance principale du cluster et l'éditeur de bloc-notes. Vous pouvez utiliser les valeurs par défaut ou personnaliser ces groupes de sécurité. Pour plus d'informations, consultez Spécification des groupes de sécurité EC2 pour les blocs-notes EMR (p. 311).
- Vous spécifiez un rôle de service AWS qui détermine les autorisations Bloc-notes EMR a lorsque vous interagissez avec d'autres AWS services. Pour plus d'informations, consultez Service Role for Blocs-notes EMR (p. 225).

Installation et utilisation des noyaux et des bibliothèques

Chaque Bloc-notes EMR est livré avec un ensemble de bibliothèques et de noyaux pré-installés. Vous pouvez installer des bibliothèques et des noyaux supplémentaires dans un cluster Amazon EMR si le cluster a accès au référentiel où se trouvent les noyaux et les bibliothèques. Par exemple, pour les clusters dans les sous-réseaux privés, vous devrez peut-être configurer la traduction d'adresses réseau (NAT) et fournir un chemin d'accès au référentiel PYPI public pour installer une bibliothèque. Pour plus d'informations sur la configuration d'un accès externe pour différentes configurations réseau, voir Scénarios et exemples dans le Amazon VPC Guide de l'utilisateur.

Installation des noyaux et des bibliothèques Python sur un nœud maître de cluster

Avec Amazon EMR version 5.30.0 et ultérieure, à l'exception de la version 6.0.0, vous pouvez installer des bibliothèques Python et des noyaux supplémentaires sur le nœud maître du cluster. Après l'installation, ces noyaux et bibliothèques sont disponibles pour tout utilisateur exécutant un Bloc-notes EMR attaché au cluster. Les bibliothèques Python installées de cette façon ne sont disponibles que pour les processus s'exécutant sur le nœud maître. Les bibliothèques ne sont pas installées sur les nœuds principaux ou de tâche et ne sont pas disponibles pour les exécuteurs s'exécutant sur ces nœuds.

Vous pouvez installer des nels et des bibliothèques en exécutant `pip` ou `conda` dans `/emr/notebook-env/bin` sur le noeud maître.

Vous pouvez exécuter `pip` en tant que commande de terminal à partir d'une cellule de bloc-notes. L'utilisation de `conda` nécessite un accès sudo. Vous devez vous connecter au nœud maître à l'aide de SSH, puis exécuter `conda` à partir du terminal. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368).

L'exemple suivant illustre l'installation `sas_kernel` utiliser une commande de terminal depuis un Bloc-notes EMR cellule utilisant le noyau Python :

```
/emr/notebook-env/bin/pip install sas_kernel
```

Utilisation de bibliothèques limitées aux bloc-notes

Les bibliothèques à portée de bloc-notes sont disponibles à l'aide de clusters créés à l'aide de Amazon EMR version 5.26.0 ou ultérieure. Les bibliothèques à portée de bloc-notes sont destinées à être utilisées uniquement avec le noyau PySpark. Tout utilisateur peut installer des bibliothèques supplémentaires à portée de bloc-notes à partir d'une cellule de bloc-notes. Ces bibliothèques ne sont disponibles que pour cet utilisateur de bloc-notes au cours d'une seule session de bloc-notes. Si d'autres utilisateurs ont besoin des mêmes bibliothèques ou si le même utilisateur a besoin des mêmes bibliothèques dans une session différente, la bibliothèque doit être réinstallée.

Considérations et restrictions

Tenez compte des éléments suivants lorsque vous utilisez des bibliothèques limitées au bloc-notes :

- Vous ne pouvez désinstaller que les bibliothèques installées à l'aide de `install_pypi_package` API. Vous ne pouvez désinstaller aucune bibliothèque installée sur le cluster.
- Si les mêmes bibliothèques avec des versions différentes sont installées sur le cluster et en tant que bibliothèques limitées au bloc-notes, la version de la bibliothèque limitée aux bloc-notes remplace la version de la bibliothèque du cluster.

Utilisation de bibliothèques limitées aux bloc-notes

Pour installer des bibliothèques, votre cluster Amazon EMR doit avoir accès au référentiel PyPI dans lequel se trouvent les bibliothèques.

Les exemples suivants montrent des commandes simples permettant de répertorier, d'installer et de désinstaller des bibliothèques à partir d'une cellule de bloc-notes à l'aide des API et du noyau PySpark. Pour des exemples supplémentaires, voir Installer des bibliothèques Python sur un cluster en cours de fonctionnement avec des carnets EMR publier sur le blog AWS Big Data.

Example – Liste des bibliothèques actuelles

La commande suivante répertorie les ensembles Python disponibles pour la session de bloc-notes Spark actuelle. Cette liste contient les bibliothèques installées sur le cluster et les bibliothèques limitées aux bloc-notes.

```
sc.list_packages()
```

Example – Installation de la bibliothèque Celery

La commande suivante installe le Céleri bibliothèque en tant que bibliothèque portée par ordinateur portable.

```
sc.install_pypi_package("celery")
```

Après avoir installé la bibliothèque, la commande suivante confirme qu'elle est disponible sur le pilote et les programmes d'exécution Spark.

```
import celery
sc.range(1,10000,1,100).map(lambda x: celery.__version__).collect()
```

Example – Installation de la bibliothèque Arrow, spécification de la version et du référentiel

La commande suivante installe le Flèche bibliothèque en tant que bibliothèque étendue par ordinateur portable, avec une spécification de la version de bibliothèque et de l'URL du référentiel.

```
sc.install_pypi_package("arrow==0.14.0", "https://pypi.org/simple")
```

Example – Désinstallation d'une bibliothèque

La commande suivante désinstalle la bibliothèque Arrow, en la supprimant en tant que bibliothèque limitées aux bloc-notes de la session en cours.

```
sc.uninstall_package("arrow")
```

Association de référentiels Git à Blocs-notes EMR

Vous pouvez associer des référentiels Git à vos blocs-notes Amazon EMR pour les enregistrer dans un environnement contrôlé par version. Vous pouvez associer jusqu'à trois référentiels à un bloc-notes. Les services Git suivants sont pris en charge :

- AWS CodeCommit
- GitHub

- Bitbucket

Associer des référentiels basés sur Git à votre bloc-notes présente les avantages suivants.

- Contrôle de version – Vous pouvez enregistrer les modifications de code dans un système de contrôle de version afin que vous puissiez examiner l'historique de vos modifications et les inverser sélectivement.
- Collaboration – Les collègues travaillant dans des carnets de notes peuvent partager le code via des référentiels à distance à Git. Les bloc-notes permettent de cloner ou fusionner du code de référentiels Git distants et de renvoyer les modifications vers ces référentiels distants.
- Réutilisation du code – De nombreux portables Jupyter qui démontrent des techniques d'analyse des données ou d'apprentissage automatique sont disponibles dans des référentiels hébergés publiquement, comme github. Vous pouvez associer vos blocs-notes à un référentiel pour réutiliser les blocs-notes Jupyter contenus dans ce référentiel.

Pour utiliser des référentiels Git avec Blocs-notes EMR, vous ajoutez les référentiels comme ressources dans le Amazon EMR la console, les informations d'identification des employés pour les référentiels qui nécessitent une authentification et les liens avec vos carnets. Vous pouvez afficher une liste de référentiels stockés dans votre compte et des détails sur chaque référentiel dans le Amazon EMR console. Vous pouvez associer un référentiel Git existant à un bloc-notes lorsque de sa création.

Rubriques

- Prérequis et considérations (p. 45)
- Ajout d'un référentiel Git à Amazon EMR (p. 46)
- Mise à jour ou suppression d'un référentiel Git (p. 47)
- Association ou dissociation d'un référentiel Git (p. 47)
- Création d'un nouveau bloc-notes avec un référentiel Git associé (p. 49)
- Utilisation de référentiels Git dans un bloc-notes (p. 49)

Prérequis et considérations

Tenez compte de ce qui suit lorsque vous prévoyez d'intégrer un référentiel Git avec Blocs-notes EMR.

AWS CodeCommit

Si vous utilisez un référentiel CodeCommit, vous devez utiliser les informations d'identification Git et HTTPS avec CodeCommit. Clés SSH et HTTPS avec AWS CLI l'assistant d'informations d'identification ne sont pas pris en charge. CodeCommit ne prend pas en charge les jetons d'accès personnels (pats). Pour plus d'informations, voir Utilisation d'IAM avec codéclassement : Informations d'identification Git, clés SSH et clés d'accès AWS dans le IAM Guide de l'utilisateur et Configuration pour les utilisateurs HTTPS utilisant les identifiants Git dans le AWS CodeCommit Guide de l'utilisateur.

Considérations relatives à l'accès et aux autorisations

Avant d'associer un référentiel à votre bloc-notes, vous devez vous assurer que votre cluster, votre rôle IAM pour Blocs-notes EMR et vos groupes de sécurité disposent des paramètres et autorisations appropriés.

- Accès à Internet requis – L'interface réseau lancée n'a qu'une adresse IP privée. Cela signifie que le cluster auquel votre bloc-notes se connecte doit se trouver dans un sous-réseau privé doté d'une passerelle NAT (Network Address Translation) ou doit pouvoir accéder à Internet via une passerelle privée virtuelle. Pour plus d'informations, voir Options VPC Amazon.

  Les groupes de sécurité de votre bloc-notes doivent inclure une règle sortante pour permettre au bloc-notes d'acheminer le trafic vers Internet depuis le cluster. Il est recommandé de créer vos propres

groupes de sécurité. Pour plus d'informations, voir Spécification des groupes de sécurité EC2 pour les carnets DME.

> **Important**
>
> Si l'interface réseau est lancée dans un sous-réseau public, elle ne pourra pas communiquer avec Internet via une passerelle Internet (IGW).

- Autorisations pour AWS Secrets Manager – Si vous utilisez Secrets Manager pour stocker les secrets que vous utilisez pour accéder à un référentiel, le the section called "Blocs-notes EMR Role" (p. 225) doit disposer d'une politique d'autorisations qui permet à la `secretsmanager:GetSecretValue`.

Ajout d'un référentiel Git à Amazon EMR

Pour ajouter un référentiel Git comme ressource dans votre Amazon EMR compte

1. Ouvrez la console Amazon EMR, à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisir Référentiels de Git, puis choisissez Ajouter un référentiel.
3. Pour Nom du référentiel, entrez un nom à utiliser pour le référentiel dans Amazon EMR.

   Les noms ne peuvent contenir que des caractères alphanumériques, des traits d'union (-) ou des traits de soulignement (_).
4. Pour URL du référentiel Git, saisissez l'URL du référentiel. Lorsque vous utilisez un CodeCommit référentiel, c'est l'URL qui est copiée lorsque vous choisissez Cloner URL puis Cloner HTTPS, par exemple, `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/MyCodeCommitRepoName`.
5. Pour Succursale, entrez un nom de succursale.
6. Pour Informations d'identification Git, choisissez les options conformément aux directives suivantes. Vous pouvez utiliser un nom d'utilisateur et un mot de passe Git ou un jeton d'accès personnel (PAT) pour vous authentifier auprès de votre référentiel. Blocs-notes EMR accède à vos informations d'identification Git en utilisant les secrets stockés dans Secrets Manager.

Option	Description :
Utilisation d'un secret AWS existant	Choisissez cette option si vous avez déjà enregistré vos informations d'identification en tant que secret dans Secrets Manager, puis sélectionnez le nom secret dans la liste. Si vous sélectionnez un secret associé à un nom d'utilisateur et un mot de passe Git, le secret doit être au format `{"gitUsername": MyUserName, "gitPassword": MyPassword}`.
Création d'un secret	Choisissez cette option pour associer les informations d'identification Git existantes à un nouveau secret dans lequel vous créerez Secrets Manager. Effectuez l'une des opérations suivantes en fonction des informations d'identification Git que vous utilisez pour le référentiel. Si vous utilisez un nom d'utilisateur et un mot de passe Git pour accéder au référentiel, sélectionnez Nom d'utilisateur et mot de passe,

Option	Description :
	entrez le Nom secret à utiliser dans Secrets Manager, puis saisissez Nom d' et Mot de passe à associer au secret. –OU– Si vous utilisez un jeton d'accès personnel pour accéder au référentiel, sélectionnez Jeton d'accès personnel (PAT), entrez le Nom secret à utiliser dans Secrets Manager, puis saisissez votre jeton d'accès personnel. Pour plus d'informations, voir Création d'un jeton d'accès personnel pour la ligne de commande pour github et Jetons d'accès personnels pour Bitbucket. CodeCommit les référentiels ne prennent pas en charge cette option.
Utilisation d'un référentiel public sans informations d'identification	Choisissez cette option pour accéder à un référentiel public.

7. Choisir Ajouter un référentiel.

Mise à jour ou suppression d'un référentiel Git

Pour mettre à jour un référentiel Git

1. Sur le Référentiels de Git , choisissez le référentiel à mettre à jour.
2. Sur la page du référentiel, choisissez Modifier le référentiel.
3. Mise à jour Informations d'identification Git sur la page du référentiel.

Pour supprimer un référentiel Git

1. Sur le Référentiels de Git , choisissez le référentiel que vous souhaitez supprimer.
2. Sur la page du référentiel, sélectionnez tous les blocs-notes actuellement liés au référentiel. Choisir Dissocier le carnet.
3. Sur la page du référentiel, choisissez Supprimer.

   Note

   Pour supprimer le référentiel Git local de Amazon EMR, vous devez d'abord annuler les liens entre les blocs-notes et ce référentiel. Pour plus d'informations, consultez Association ou dissociation d'un référentiel Git (p. 47). La suppression d'un référentiel Git ne supprime aucun secret créé pour le référentiel. Vous pouvez supprimer le secret dans AWS Secrets Manager.

Association ou dissociation d'un référentiel Git

Pour associer un référentiel Git à un bloc-notes EMR

Le référentiel peut être relié à un carnet une fois que le carnet est Prêt.

1. Du Carnets , choisissez le carnet que vous souhaitez mettre à jour.
2. Dans le Référentiels de Git section sur le Carnet page, choisir Lier un nouveau référentiel.

3. Dans la liste de référentiel du Lier le référentiel Git au carnet sélectionnez un ou plusieurs référentiels que vous souhaitez lier à votre carnet, puis choisissez Référentiel de liens.

Ou :

1. Sur le Référentiels de Git , choisissez le référentiel que vous souhaitez lier à votre ordinateur portable.
2. Dans la liste de Carnets DME, choisissez Lier un nouveau carnet pour lier ce référentiel à un ordinateur portable existant.

Pour annuler le lien entre un référentiel Git et un bloc-notes EMR

1. Du Carnets , choisissez le carnet que vous souhaitez mettre à jour.
2. Dans la liste de Référentiels de Git, sélectionnez le référentiel que vous souhaitez dissocier de votre carnet, puis choisissez Dissocier le référentiel.

Ou :

1. Sur le Référentiels de Git page, choisissez le référentiel à mettre à jour pour.
2. Dans la liste de Carnets DME, sélectionnez le carnet que vous souhaitez dissocier du référentiel, puis choisissez Dissocier le carnet.

> **Note**
>
> La création d'un lien entre un référentiel Git et un bloc-notes clone le référentiel distant sur votre bloc-notes Jupyter local. L'annulation d'un lien entre le référentiel Git et un bloc-notes déconnecte uniquement le bloc-notes du référentiel distant sans supprimer le référentiel Git local.

Présentation de l'état du référentiel

Un référentiel Git peut avoir l'un des statuts suivants dans la liste des référentiels. Pour de plus amples informations sur la liaison de blocs-notes EMR avec des référentiels Git, veuillez consulter Association ou dissociation d'un référentiel Git (p. 47).

Statut	Signification
Linking (Liaison en cours)	Le référentiel Git est en train d'être lié au bloc-notes. Pendant que le référentiel est Liaison, vous ne pouvez pas arrêter le carnet.
Linked (Lié)	Le référentiel Git est lié au bloc-notes. Tandis que le référentiel a Lié le statut est connecté au référentiel distant.
Link Failed (Échec du lien)	Le référentiel Git n'a pas pu se lier au bloc-notes. Vous pouvez réessayer.
Unlinking (Annulation du lien en cours)	Le lien entre le référentiel Git et le bloc-notes est en cours d'annulation. Pendant que le référentiel est Dissocier, vous ne pouvez pas arrêter le carnet. L'annulation d'un lien entre un référentiel Git et un bloc-notes déconnecte uniquement le bloc-notes du référentiel distant sans supprimer de code du bloc-notes.

Statut	Signification
Unlink Failed (Échec de l'annulation du lien)	L'annulation du lien entre le référentiel Git et le bloc-notes a échoué. Vous pouvez réessayer.

Création d'un nouveau bloc-notes avec un référentiel Git associé

Pour créer un bloc-notes et l'associer à des référentiels Git dans l'AWS Management Console

1. Suivez les instructions décrites dans la section Création d'un bloc-notes (p. 27).
2. Pour Groupe de sécurité, choisissez Utilisez votre propre groupe de sécurité.

   Note

   Les groupes de sécurité de votre bloc-notes doivent inclure une règle sortante pour permettre au bloc-notes d'acheminer le trafic vers Internet via le cluster. Il est recommandé de créer vos propres groupes de sécurité. Pour plus d'informations, voir Spécification des groupes de sécurité EC2 pour les carnets DME.
3. Pour Référentiels de Git, Choisir un référentiel à associer au carnet.

   1. Choisissez un référentiel stocké comme ressource dans votre compte, puis choisissez Enregistrer.
   2. Pour ajouter un référentiel comme ressource dans votre compte, choisissez ajouter un nouveau référentiel. Complétez le Ajouter un référentiel dans une nouvelle fenêtre.

Utilisation de référentiels Git dans un bloc-notes

Vous pouvez choisir Ouvert à jupyterlab ou Ouvert à Jupyter lorsque vous ouvrez un ordinateur portable.

Si vous choisissez d'ouvrir le bloc-notes dans Jupyter, une liste de fichiers et de dossiers extensibles dans le bloc-notes s'affiche. Vous pouvez exécuter manuellement des commandes Git comme ci-après dans une cellule de bloc-notes.

```
!git pull origin master
```

Pour ouvrir un des référentiels supplémentaires, accédez à d'autres dossiers.

Si vous choisissez d'ouvrir le bloc-notes avec une interface JupyterLab, l'extension jupyter-git est installée et disponible pour l'utilisation. Pour plus d'informations sur l'extension jupyter-git pour jupyterlab, voir jupyterlab-git.

Plan and Configure Clusters

Cette section explique les options et les instructions de configuration pour la planification, la configuration et le lancement des clusters avec Amazon EMR. Avant de lancer un cluster, vous effectuez des choix concernant votre système en fonction des données que vous traitez et de vos exigences en termes de coût, de vitesse, de capacité, de disponibilité, de sécurité et de capacité de gestion. Vos choix incluent :

- What region to run a cluster in, where and how to store data, and how to output results. See Configure Cluster Location and Data Storage (p. 50).
- Whether you are running Amazon EMR clusters on Outposts or Zones locales. See Clusters EMR sur AWS Outposts (p. 77) or Clusters EMR dans les Zones locales AWS (p. 79).
- Whether a cluster is long-running or transient, and what software it runs. See Configuring a Cluster to Auto-Terminate or Continue (p. 110) and Configure Cluster Software (p. 125).
- Whether a cluster has a single master node or three master nodes. See Planification et configuration des nœuds maîtres (p. 68).
- The hardware and networking options that optimize cost, performance, and availability for your application. See Configuration du matériel et de la mise en réseau d'un cluster (p. 129).
- How to set up clusters so you can manage them more easily, and monitor activity, performance, and health. See Configure Cluster Logging and Debugging (p. 166) and Tag Clusters (p. 171).
- How to authenticate and authorize access to cluster resources, and how to encrypt data. See Sécurité dans Amazon EMR (p. 177).
- How to integrate with other software and services. See Drivers and Third-Party Application Integration (p. 176).

Configure Cluster Location and Data Storage

Cette section décrit comment configurer la région d'un cluster, les différents systèmes de fichier disponibles lorsque vous utilisez Amazon EMR et leur utilisation. Elle couvre également la préparation ou le téléchargement des données vers Amazon EMR si nécessaire, ainsi que le mode de préparation d'un emplacement de sortie pour les fichiers journaux et les fichiers de données de sortie que vous configurez.

Rubriques

- Choose an AWS Region (p. 50)
- Work with Storage and File Systems (p. 52)
- Prepare Input Data (p. 54)
- Configure an Output Location (p. 63)

Choose an AWS Region

Amazon Web Services s'exécute sur des serveurs dans des centres de données répartis dans le monde entier. Ces centres de données sont organisés par région géographique. Lorsque vous lancez un cluster Amazon EMR, vous devez spécifier une région. Vous pouvez choisir une région pour réduire la latence, minimiser les coûts ou répondre à des exigences réglementaires. Pour obtenir la liste des régions et points de terminaison pris en charge par Amazon EMR, consultez Régions et points de terminaison dans le Référence générale d'Amazon Web Services.

Pour de meilleures performances, vous devez lancer le cluster dans la région où se trouvent vos données. Par exemple, si le compartiment Amazon S3 qui stocke vos données d'entrée se trouve dans la région USA Ouest (Oregon), vous devez lancer votre cluster dans la région USA Ouest (Oregon) pour éviter les frais de transfert de données entre régions. Si vous utilisez un compartiment Amazon S3 pour recevoir les données de sortie du cluster, vous pouvez également le créer dans la région USA Ouest (Oregon).

Si vous envisagez d'associer une paire de clés Amazon EC2 au cluster (requis pour utiliser SSH pour vous connecter au nœud maître), la paire de clés doit être créée dans la même région que le cluster. De même, les groupes de sécurité qu'Amazon EMR crée pour gérer le cluster sont créés dans la même région que le cluster.

Si vous êtes inscrit à un compte AWS depuis le 17 mai 2017, la région par défaut lorsque vous accédez à une ressource à partir de la AWS Management Console est USA Est (Ohio) (us-east-2) ; pour les comptes antérieurs, la région par défaut est soit USA Ouest (Oregon) (us-west-2), soit USA Est (Virginie du Nord) (us-east-1). Pour plus d'informations, consultez Régions et points de terminaison.

Certaines fonctionnalités AWS sont disponibles uniquement dans des régions limitées. Par exemple, les instances Cluster Compute sont disponibles uniquement dans la région USA Est (Virginie du Nord) et la région Asie-Pacifique (Sydney) prend en charge uniquement Hadoop 1.0.3 ou une version ultérieure. Lorsque vous choisissez une région, vérifiez qu'elle prend en charge les fonctionnalités que vous voulez utiliser.

Pour de meilleures performances, utilisez la même région pour toutes vos ressources AWS qui seront utilisées avec le cluster. Le tableau suivant met en correspondance les noms des régions entre les services. Pour connaître la liste des régions Amazon EMR, consultez Régions et points de terminaison AWS dans le Référence générale d'Amazon Web Services.

Choose a Region Using the Console

Votre région par défaut s'affiche automatiquement.

Pour modifier les régions à l'aide de la console

- Pour changer de région, choisissez la liste des régions à droite de vos informations de compte dans la barre de navigation.

Specify a Region Using the AWS CLI

Vous spécifiez une région par défaut dans l'AWS CLI en utilisant soit la commande aws configure ou la variable d'environnement `AWS_DEFAULT_REGION`. Pour plus d'informations, consultez Configuration de la région AWS dans le AWS Command Line Interface Guide de l'utilisateur.

Choose a Region Using an SDK or the API

Pour choisir une région à l'aide d'un kit SDK, configurez votre application pour utiliser le point de terminaison de cette région. Si vous créez une application cliente à l'aide d'un kit AWS SDK, vous pouvez changer le point de terminaison client en appelant `setEndpoint`, comme illustré dans l'exemple suivant :

```
client.setEndpoint("elasticmapreduce.us-west-2.amazonaws.com");
```

Une fois que votre application a spécifié une région en définissant le point de terminaison, vous pouvez définir la zone de disponibilité pour les instances EC2 de votre cluster. Les zones de disponibilité sont des emplacements géographiques distincts qui sont conçus pour être isolés des défaillances dans d'autres zones de disponibilité et fournir une connectivité réseau à faible latence et peu onéreuse aux autres zones de disponibilité dans la même région. Une région est constituée d'une ou de plusieurs zones de

disponibilité. Pour optimiser les performances et réduire la latence, toutes les ressources doivent être situées dans la même zone de disponibilité que le cluster qui les utilise.

Work with Storage and File Systems

Amazon EMR et Hadoop fournissent divers systèmes de fichiers que vous pouvez utiliser lors du traitement des étapes de cluster. Vous spécifiez le système de fichiers à utiliser par le préfixe de l'URI utilisé pour accéder aux données. Par exemple, `s3://DOC-EXAMPLE-BUCKET1/path` fait référence à un compartiment Amazon S3 utilisant EMRFS. Le tableau suivant répertorie les systèmes de fichiers disponibles, avec des recommandations sur les moments où il est préférable de les utiliser.

Amazon EMR et Hadoop utilisent généralement deux des systèmes de fichiers suivants ou plus lors du traitement d'un cluster. HDFS et EMRFS sont les deux systèmes de fichiers principaux utilisés avec Amazon EMR.

> **Important**
>
> À partir de la version 5.22.0 d'Amazon EMR, Amazon EMR utilise AWS Signature version 4 exclusivement pour authentifier les demandes adressées à Amazon S3. Les versions antérieures d'Amazon EMR utilisent AWS Signature version 2 dans certains cas, sauf si les notes de mise à jour indiquent que Signature version 4 est utilisé exclusivement. Pour de plus amples informations, veuillez consulter Authentification des demandes (AWS Signature version 4) et Authentification des demandes (AWS Signature version 2) dans le Manuel du développeur Amazon Simple Storage Service.

Système de fichiers	Préfixe	Description :
HDFS	`hdfs://` (ou aucun préfixe)	HDFS est un système de fichiers distribué, évolutif et portable pour Hadoop. L'un des avantages de HDFS est la reconnaissance des données entre les nœuds de cluster Hadoop qui gèrent les clusters et les nœuds de cluster Hadoop qui gèrent les étapes individuelles. Pour plus d'informations, consultez la documentation Hadoop. HDFS est utilisé par les nœuds maîtres et principaux. Il présente l'avantage d'être rapide, mais l'inconvénient d'être un stockage éphémère qui est récupéré lorsque le cluster se termine. Il est particulièrement adapté à la mise en cache des résultats obtenus aux étapes de flux de travail intermédiaires.
EMRFS	`s3://`	EMRFS est une implémentation du système de fichiers Hadoop, utilisée pour lire et écrire des fichiers standard d'Amazon EMR directement sur Amazon S3. EMRFS permet de stocker des données persistantes dans Amazon S3 en vue de les utiliser avec Hadoop, tout en fournissant des fonctionnalités telles que le chiffrement côté serveur Amazon S3, la cohérence en lecture après écriture et la cohérence des listes. **Note** Précédemment, Amazon EMR utilisé le système de fichiers natif S3 avec le schéma URI, `s3n`. Bien que cela fonctionne toujours, nous vous recommandons d'utiliser `s3` Le système URI pour les meilleures performances, la meilleure sécurité et la meilleure fiabilité.

Système de fichiers	Préfixe	Description :
système de fichiers local		Le système de fichiers local fait référence à un disque connecté localement. Lorsqu'un cluster Hadoop est créé, chaque noeud est créé à partir d'une instance EC2 qui est livrée avec un bloc préconfiguré de stockage de disque préconnecté appelé instance store. Les données sur les volumes de magasins d'instance ne persistent que pendant la durée de vie de son instance EC2. Les volumes de stockage d'instance conviennent parfaitement pour le stockage de données temporaires qui changent en permanence, telles que les tampons, les caches, les données de travail et d'autres contenus temporaires. Pour plus d'informations, consultez Stockage d'instance Amazon EC2.
système de fichiers à blocs Amazon S3 (hérité)	`s3bfs://`	Le système de fichiers à blocs Amazon S3 est un système de stockage de fichiers hérité. Nous déconseillons vivement l'utilisation de ce système. **Important** Nous vous recommandons de ne pas utiliser ce système de fichiers, car il peut déclencher une condition de concurrence susceptible d'entraîner l'échec de votre cluster. Toutefois, il peut être requis par des applications héritées.

Note

Le protocole `s3a` n'est pas pris en charge. Nous vous suggérons d'utiliser `s3` à la place de `s3a`.

Access File Systems

Vous spécifiez le système de fichiers à utiliser par le préfixe de l'identifiant de ressource uniforme (URI) utilisé pour accéder aux données. Les procédures suivantes montrent comment faire référence à plusieurs types de système de fichiers.

Pour accéder à un système HDFS local

- Spécifiez le préfixe `hdfs:///` dans l'URI. Amazon EMR résout les chemins d'accès qui ne spécifient pas de préfixe dans l'URI pour le système HDFS local. Par exemple, les deux URI suivants font référence au même emplacement dans HDFS.

```
hdfs:///path-to-data

/path-to-data
```

Pour accéder à un système HDFS distant

- Incluez l'adresse IP du nœud maître dans l'URI, comme illustré dans les exemples suivants.

```
hdfs://master-ip-address/path-to-data
```

```
master-ip-address/path-to-data
```

Pour accéder à Amazon S3

- Utilisez le préfixe `s3://`.

```
s3://bucket-name/path-to-file-in-bucket
```

Pour accéder au système de fichiers à blocs Amazon S3

- Utilisez-le uniquement pour les applications héritées qui requièrent le système de fichiers à blocs Amazon S3. Pour accéder ou stocker des données avec ce système de fichiers, utilisez le préfixe `s3bfs://` dans l'URI.

  Le système de fichiers à blocs Amazon S3 est un système de fichiers hérité qui était utilisé pour prendre en charge les chargements vers Amazon S3 dont la taille dépassait 5 Go. Avec la fonctionnalité de téléchargement partitionné qu'Amazon EMR fournit via le kit AWS SDK pour Java, vous pouvez charger des fichiers d'une taille pouvant atteindre jusqu'à 5 To vers le système de fichiers natif de Amazon S3 et le système de fichiers à blocs Amazon S3 est obsolète.

  Warning

  Etant donné que ce système de fichiers hérité peut créer des conditions de concurrence susceptibles d'endommager le système de fichiers, vous devez éviter ce format et utiliser EMRFS à la place.

```
s3bfs://bucket-name/path-to-file-in-bucket
```

Prepare Input Data

La plupart des clusters chargent les données d'entrée, puis traitent ces données. Pour pouvoir être chargées, les données doivent être dans un emplacement auquel le cluster peut accéder et dans un format que le cluster peut traiter. Le scénario le plus commun est de télécharger des données d'entrée dans Amazon S3. Amazon EMR fournit les outils permettant à votre cluster d'importer ou de lire des données de Amazon S3.

Le format d'entrée par défaut dans Hadoop correspond à des fichiers texte, mais vous pouvez personnaliser Hadoop et utiliser des outils pour importer des données stockées dans d'autres formats.

Rubriques

- Types of Input Amazon EMR Can Accept (p. 54)
- How to Get Data Into Amazon EMR (p. 55)

Types of Input Amazon EMR Can Accept

Le format d'entrée par défaut pour un cluster correspond à des fichiers texte dont chaque ligne est séparée par un caractère de nouvelle ligne (\n), ce qui est le format d'entrée le plus couramment utilisé.

Si vos données d'entrée sont dans un format différent des fichiers texte par défaut, vous pouvez utiliser l'interface Hadoop `InputFormat` pour spécifier d'autres types d'entrée. Vous pouvez même créer une sous-classe de la classe `FileInputFormat` pour gérer les types de données personnalisés. Pour plus d'informations, consultez http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/InputFormat.html.

Si vous utilisez Hive, vous pouvez utiliser un sérialiseur/désérialiseur (SerDe) pour lire les données à partir d'un format donné dans HDFS. Pour plus d'informations, consultez https://cwiki.apache.org/confluence/display/Hive/SerDe.

How to Get Data Into Amazon EMR

Amazon EMR fournit plusieurs méthodes pour introduire des données dans un cluster. La méthode la plus courante consiste à charger les données vers Amazon S3 et à utiliser les fonctionnalités intégrées d'Amazon EMR pour charger les données sur votre cluster. Vous pouvez également utiliser la fonctionnalité de cache distribué de Hadoop pour transférer des fichiers à partir d'un système de fichiers distribué vers le système de fichiers local. L'implémentation de Hive fournie par Amazon EMR (Hive version 0.7.1.1 ou ultérieure) inclut des fonctionnalités que vous pouvez utiliser pour importer et exporter des données entre DynamoDB et un cluster Amazon EMR. Si vous avez de grandes quantités de données sur site à traiter, le service AWS Direct Connect s'avèrera vraisemblablement utile.

Rubriques

- Upload Data to Amazon S3 (p. 55)
- Import files with Distributed Cache (p. 59)
- How to Process Compressed Files (p. 62)
- Import DynamoDB Data into Hive (p. 63)
- Connect to Data with AWS DirectConnect (p. 63)
- Upload Large Amounts of Data with AWS Import/Export (p. 63)

Upload Data to Amazon S3

Pour plus d'informations sur la façon de télécharger des objets vers Amazon S3, voir Ajouter un objet à votre seau dans le Amazon Simple Storage Service Guide de mise en route. Pour plus d'informations sur l'utilisation Amazon S3 avec Hadoop, voir http://wiki.apache.org/hadoop/AmazonS3.

Rubriques

- Create and Configure an Amazon S3 Bucket (p. 55)
- Configure Multipart Upload for Amazon S3 (p. 56)
- Best Practices (p. 58)

Create and Configure an Amazon S3 Bucket

Amazon EMR utilise le AWS SDK for Java avec Amazon S3 pour stocker des données d'entrée, des fichiers journaux et des données de sortie. Amazon S3 désigne ces emplacements de stockage buckets. Les seaux ont certaines restrictions et limitations pour se conformer à Amazon S3 et DNS. Pour plus d'informations, consultez Limites et restrictions applicables aux compartiments dans le manuel Amazon Simple Storage Service Manuel du développeur.

Cette section vous montre comment utiliser la AWS Management Console Amazon S3 pour créer et définir des autorisations pour un compartiment Amazon S3. Vous pouvez également créer et définir des autorisations pour un compartiment Amazon S3 à l'aide de l'API Amazon S3 ou de l'AWS CLI. Vous pouvez aussi utiliser Curl avec une modification pour transmettre les paramètres d'authentification appropriés pour Amazon S3.

Consultez les ressources suivantes.

- To create a bucket using the console, see Create a Bucket in the Amazon Simple Storage Service Guide de l'utilisateur de la console.
- To create and work with buckets using the AWS CLI, see Using High-Level S3 Commands with the AWS Command Line Interface in the Amazon Simple Storage Service Guide de l'utilisateur de la console.
- To create a bucket using an SDK, see Examples of Creating a Bucket in the Amazon Simple Storage Service Manuel du développeur.
- To work with buckets using Curl, see Amazon S3 Authentication Tool for Curl.
- For more information on specifying Region-specific buckets, see Accessing a Bucket in the Amazon Simple Storage Service Manuel du développeur.

> **Note**
>
> Si vous activez la journalisation pour un compartiment, seuls les journaux d'accès du compartiment sont activés, pas les journaux du cluster Amazon EMR.

Pendant ou après la création du compartiment, vous pouvez définir les autorisations appropriées pour accéder au compartiment en fonction de votre application. Habituellement, vous (le propriétaire) vous donnez accès en lecture et en écriture et accordez l'accès en lecture aux utilisateurs authentifiés.

Les compartiments Amazon S3 requis doivent avoir été créés pour que vous puissiez créer un cluster. Vous devez charger les scripts obligatoires ou les données référencées dans le cluster vers Amazon S3. Le tableau suivant décrit des exemples de données, de scripts et d'emplacements de fichier journal.

Configure Multipart Upload for Amazon S3

Amazon EMR prend en charge le téléchargement partitionné Amazon S3 via le kit SDK AWS pour Java. Le téléchargement partitionné vous permet de charger un objet unique sous la forme d'un ensemble de parties. Vous pouvez charger ces parties d'objet indépendamment et dans n'importe quel ordre. Si le transfert d'une partie échoue, vous pouvez la retransférer sans affecter les autres. Une fois le chargement de toutes les parties de l'objet terminé, Amazon S3 les assemble et crée l'objet.

Pour plus d'informations, consultez Présentation du chargement partitionné dans le Amazon Simple Storage Service Manuel du développeur.

De plus, Amazon EMR propose des propriétés qui vous permettent de contrôler de manière plus précise le nettoyage des parties de chargements partitionnés échoués.

Le tableau suivant décrit les propriétés de configuration Amazon EMR pour un chargement partitionné. Vous pouvez configurer ces éléments à l'aide de la classification de configuration `core-site`. Pour plus d'informations, consultez Configuration des applications dans le Amazon EMR Guide de version.

Nom de paramètre de configuration	Default Value	Description :
`fs.s3n.multipart.uploads.enabled`	`true`	Type booléen qui indique s'il convient d'activer les téléchargements partitionnés. Lorsqu'EMRFS Vue cohérente (p. 84) est activé, les chargements partitionnés sont activés par défaut et le paramétrage de cette valeur à `false` est ignoré.
`fs.s3n.multipart.uploads.split.size`	`134217728`	Spécifie la taille maximale d'une partie, en octets, avant qu'EMRFS lance un nouveau

Nom de paramètre de configuration	Default Value	Description :
		chargement de partie lorsque le chargement partitionné est activé. La valeur minimale est `5242880` (5 Mo). Si une valeur plus faible est spécifiée, `5242880` est utilisé. Le maximum est `5368709120` (5 Go). Si une valeur supérieure est spécifiée, `5368709120` est utilisé. Si le chiffrement EMRFS côté client est désactivé et que le Valideur optimisé Amazon S3 est également désactivé, cette valeur contrôle également la dimension maximale qu'un fichier de données peut atteindre jusqu'à ce qu'EMRFS utilise le chargement partitionné plutôt qu'une demande `PutObject` pour charger le fichier. Pour plus d'informations, consultez ,
`fs.s3n.ssl.enabled`	`true`	Type booléen qui indique s'il convient d'utiliser http ou https.
`fs.s3.buckets.create.enabled`	`false`	Un type booléen qui indique si un compartiment devrait être créé s'il n'existe pas. Le réglage de `false` entraîne une exception aux opérations `CreateBucket`.
`fs.s3.multipart.clean.enabled`	`false`	Type booléen qui indique s'il convient d'activer le nettoyage périodique en arrière plan des chargements partitionnés inachevés.
`fs.s3.multipart.clean.age.threshold`	`604800`	Un type long qui spécifie l'âge minimal d'un chargement partitionné, en secondes, avant qu'il ne soit affecté à un nettoyage. La valeur par défaut est une semaine.
`fs.s3.multipart.clean.jitter.max`	`10000`	Un type de nombre entier qui spécifie la quantité maximale de délai de sautillement aléatoire en secondes ajouté au délai de 15 minutes fixe avant de programmer le prochain nettoyage.

Disable Multipart Upload Using the Amazon EMR Console

Cette procédure explique comment désactiver le chargement partitionné à l'aide de la console Amazon EMR lorsque vous créez un cluster.

Pour désactiver les chargements partitionnés.

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Sous Edit Software Settings (Modifier les paramètres de logiciels) et saisissez la configuration suivante : `classification=core-site,properties=[fs.s3.multipart.uploads.enabled=false]`
4. Procédez à la création du cluster.

Disable Multipart Upload Using the AWS CLI

Cette procédure explique comment désactiver le téléchargement partitionné à l'aide de l'AWS CLI. Pour désactiver le téléchargement partitionné, tapez la commande `create-cluster` avec le paramètre `--bootstrap-actions`.

Pour désactiver le téléchargement partitionné à l'aide de l'AWS CLI

1. Créez un dossier, `myConfig.json` avec le contenu suivant et enregistrez-le dans le même répertoire où vous exécutez la commande :

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3n.multipart.uploads.enabled": "false"
    }
  }
]
```

2. Tapez la commande suivante et remplacez *myKey* avec le nom de votre paire de clés EC2.

   Note

   Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-5.30.1 --applications Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --configurations file://myConfig.json
```

Disable Multipart Upload Using the API

Pour de plus amples informations sur l'utilisation des téléchargements partitionnés Amazon S3 par programmation, veuillez consulter Utilisation du kit SDK AWS pour Java pour le chargement partitionné dans le Amazon Simple Storage Service Manuel du développeur.

Pour de plus amples informations sur les kits SDK AWS pour Java, veuillez consulter Kits SDK AWS pour Java.

Best Practices

Vous trouverez ci-dessous des recommandations pour l'utilisation des compartiments Amazon S3 avec les clusters EMR.

Enable Versioning

La gestion des versions est une configuration recommandée pour votre compartiment Amazon S3. En activant la gestion des versions, vous vous assurez que si des données sont supprimées ou remplacées accidentellement, elles peuvent être récupérées. Pour plus d'informations, consultez la page Gestion des versions dans le Amazon Simple Storage Service Manuel du développeur.

Clean Up Failed Multipart Uploads

Les composants des clusters EMR utilisent des chargements partitionnés via le kit AWS SDK pour Java avec les API Amazon S3 pour écrire des fichiers journaux et des données de sortie dans Amazon S3

par défaut. Pour en savoir plus sur la modification des propriétés liées à cette configuration qui utilise Amazon EMR, consultez Configure Multipart Upload for Amazon S3 (p. 56). Parfois, le chargement d'un fichier volumineux peut se traduire par un téléchargement partitionné Amazon S3 incomplet. Lorsqu'un téléchargement partitionné ne peut pas se terminer avec succès, le téléchargement partitionné en cours continue d'occuper votre compartiment et entraîne des frais de stockage. Pour éviter un stockage excessif de fichiers, nous vous recommandons les options suivantes :

- For buckets that you use with Amazon EMR, use a lifecycle configuration rule in Amazon S3 to remove incomplete multipart uploads three days after the upload initiation date. Lifecycle configuration rules allow you to control the storage class and lifetime of objects. For more information, see Object Lifecycle Management, and Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy.
- Enable Amazon EMR's multipart cleanup feature by setting `fs.s3.multipart.clean.enabled` to `TRUE` and tuning other cleanup parameters. This feature is useful at high volume, large scale, and with clusters that have limited uptime. In this case, the `DaysAfterIntitiation` parameter of a lifecycle configuration rule may be too long, even if set to its minimum, causing spikes in Amazon S3 storage. Amazon EMR's multipart cleanup allows more precise control. For more information, see Configure Multipart Upload for Amazon S3 (p. 56).

Manage Version Markers

Nous vous recommandons d'activer une règle de configuration de cycle de vie dans Amazon S3 pour supprimer les marqueurs de suppression des objets expirés pour les compartiments que vous utilisez avec Amazon EMR. Lorsque vous supprimez un objet dans un compartiment dont les versions sont gérées, un marqueur de suppression est créé. Si toutes les versions précédentes de l'objet expirent par la suite, un marqueur de suppression d'objet expiré est conservé dans le compartiment. Aucun frais ne s'applique pour ces marqueurs de suppression, mais la suppression des marqueurs de suppression expirés peut améliorer les performances des demandes LIST. Pour plus d'informations, consultez la section Configuration du cycle de vie pour un compartiment avec gestion des versions du Amazon Simple Storage Service Guide de l'utilisateur de la console.

Performance best practices

Selon vos charges de travail, certains types d'utilisation des applications et des clusters EMR sur ces clusters peuvent se traduire par un grand nombre de demandes adressées à un compartiment. Pour plus d'informations, consultez Considérations en matière de débit de demandes et de performances dans le Amazon Simple Storage Service Manuel du développeur.

Import files with Distributed Cache

Rubriques

- Supported File Types (p. 60)
- Location of Cached Files (p. 60)
- Access Cached Files From Streaming Applications (p. 60)
- Access Cached Files From Streaming Applications Using the Amazon EMR Console (p. 60)
- Access Cached Files From Streaming Applications Using the AWS CLI (p. 61)

Le cache distribué est une fonctionnalité Hadoop qui peut améliorer l'efficacité lorsqu'une tâche de mappage ou de réduction a besoin d'accéder aux données courantes. Si votre cluster dépend d'applications ou de fichiers binaires qui n'ont pas été installés lors de la création du cluster, vous pouvez utiliser le cache distribué pour importer ces fichiers. Cette fonctionnalité permet à un nœud de cluster de lire les fichiers importés à partir de son système de fichiers local, au lieu de récupérer les fichiers à partir d'autres nœuds de cluster.

Pour plus d'informations, accédez à http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/filecache/DistributedCache.html.

Vous appelez le cache distribué lorsque vous créez le cluster. Les fichiers sont mis en cache juste avant de démarrer le travail Hadoop et les fichiers restent en cache pendant toute la durée du travail. Vous pouvez mettre en cache des fichiers stockés sur n'importe quel système de fichiers compatible Hadoop, par exemple HDFS ou Amazon S3. La taille par défaut du cache des fichiers est de 10 Go. Pour modifier la taille du cache, reconfigurez le paramètre Hadoop `local.cache.size` à l'aide de l'action d'amorçage. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125),

Supported File Types

Le cache distribué autorise les fichiers individuels et les archives. Les fichiers individuels sont mis en cache en lecture seule. Les fichiers exécutables et les fichiers binaires disposent d'autorisations d'exécution définies.

Les archives sont un ou plusieurs fichiers emballés à l'aide d'un utilitaire, comme `gzip`. Distributed Cache transmet les fichiers compressés à chaque noeud principal et décompresse l'archive dans le cadre de la mise en cache. Le cache distribué prend en charge les formats de compression suivants :

- zip
- tgz
- tar.gz
- tar
- jar

Location of Cached Files

Le cache distribué copie les fichiers uniquement vers les nœuds principaux. S'il n'y a pas de nœuds principaux dans le cluster, le cache distribué copie les fichiers vers le nœud principal.

Le cache distribué associe les fichiers de cache au répertoire de travail actuel du mappeur et du réducteur à l'aide de liens symboliques. Un lien symbolique est un alias d'emplacement de fichier et non pas l'emplacement de fichier réel. La valeur du paramètre, `yarn.nodemanager.local-dirs` dans `yarn-site.xml`, spécifie l'emplacement des fichiers temporaires. Amazon EMR définit ce paramètre sur `/mnt/mapred`, ou une variation basée sur le type d'instance et la version EMR. Par exemple, un paramètre peut avoir `/mnt/mapred` et `/mnt1/mapred`, car le type d'instance possède deux volumes éphémères. Les fichiers de cache sont situés dans un sous-répertoire de l'emplacement de fichier temporaire à l'adresse `/mnt/mapred/taskTracker/archive`.

Si vous mettez en cache un fichier individuel, le cache distribué place le fichier dans le répertoire `archive`. Si vous mettez en cache une archive, le cache distribué décompresse le fichier, crée un sous-répertoire dans `/archive` avec le même nom que le nom du fichier d'archive. Les fichiers individuels se trouvent dans le nouveau sous-répertoire.

Vous pouvez utiliser le cache distribué uniquement lorsque vous utilisez le streaming.

Access Cached Files From Streaming Applications

Pour accéder aux fichiers de cache à partir de vos applications de mappeur ou de réducteur, assurez-vous que vous avez ajouté le répertoire actif actuel (./) dans votre chemin d'application et que vous avez référencé les fichiers de cache comme s'ils étaient présents dans le répertoire actif actuel.

Access Cached Files From Streaming Applications Using the Amazon EMR Console

Vous pouvez utiliser la console Amazon EMR pour créer des clusters qui utilisent le cache distribué.

Pour spécifier les fichiers de cache distribué à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.

2. Choisissez Créer un cluster.
3. Choisissez Exécution d'étape comme mode de lancement.
4. Dans la section Étapes, dans le champ Ajouter une étape, choisissez Programme de streaming dans la liste et cliquez sur Configurer et ajouter.
5. Dans le champ Arguments, incluez les fichiers et les archives à enregistrer dans le cache et cliquez sur Ajouter.

   La taille du fichier (ou la taille totale des fichiers dans un fichier d'archives) doit être inférieure à la taille de cache allouée.

Si vous souhaitez.	Action :	Exemple	
Ajouter un fichier individuel au cache distribué	Spécifiez `-cacheFile` suivi du nom et de l'emplacement du fichier, du signe dièse (#), puis du nom que vous voulez donner au fichier lorsqu'il est placé dans le cache local.	`-cacheFile \ s3://bucket_name/file_name#cache_file_name`	
Ajouter un fichier d'archive au cache distribué	Saisissez `-cacheArchive` suivi de l'emplacement des fichiers dans Amazon S3, du signe dièse (#), puis du nom que vous voulez donner à l'ensemble des fichiers dans le cache local.	`-cacheArchive \ s3://bucket_name/ archive_name#cache_archive_name`	

6. Procédez à la configuration et au lancement de votre cluster. Votre cluster copie les fichiers vers l'emplacement du cache avant de traiter les éventuelles étapes de cluster.

Access Cached Files From Streaming Applications Using the AWS CLI

Vous pouvez utiliser l'interface de ligne de commande pour créer des clusters qui utilisent le cache distribué.

Pour spécifier les fichiers de cache distribué à l'aide de l'AWS CLI

- Pour soumettre une étape de streaming lorsqu'un cluster est créé, tapez la commande `create-cluster` avec le paramètre `--steps`. Pour spécifier les fichiers de cache distribué à l'aide de l'AWS CLI, spécifiez les arguments appropriés lorsque vous soumettez une étape de streaming.

Si vous souhaitez...	Ajoutez le paramètre suivant au cluster ...
ajouter un fichier individuel au cache distribué	Spécifiez `-cacheFile` suivi du nom et de l'emplacement du fichier, du signe dièse (#), puis du nom que vous voulez donner au fichier lorsqu'il est placé dans le cache local.
ajouter un fichier d'archive au cache distribué	saisissez `-cacheArchive` suivi de l'emplacement des fichiers dans Amazon S3, du signe dièse (#), puis du nom que vous voulez donner à l'ensemble des fichiers dans le cache local.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Example 1

Tapez la commande suivante pour lancer un cluster et soumettre une étape de streaming qui utilise `-cacheFile` pour ajouter un fichier, `sample_dataset_cached.dat`, dans le cache.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py s3://
my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py,"-input","s3://
my_bucket/my_input","-output","s3://my_bucket/my_output", "-cacheFile","s3://my_bucket/
sample_dataset.dat#sample_dataset_cached.dat"]
```

Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Example 2.

La commande suivante illustre la création d'un cluster de streaming et utilise `-cacheArchive` pour ajouter une archive de fichiers dans le cache.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py s3://
my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py,"-input","s3://
my_bucket/my_input","-output","s3://my_bucket/my_output", "-cacheArchive","s3://my_bucket/
sample_dataset.tgz#sample_dataset_cached"]
```

Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

How to Process Compressed Files

Hadoop vérifie l'extension de fichier pour détecter les fichiers compressés. Les types de compression pris en charge par Hadoop sont : gzip, bzip2 et LZO. Vous n'avez pas besoin d'entreprendre d'action supplémentaire pour extraire les fichiers à l'aide de ces types de compression ; Hadoop s'en occupe pour vous.

Pour indexer les fichiers LZO, vous pouvez utiliser la bibliothèque hadoop-lzo qui peut être téléchargée à partir de https://github.com/kevinweil/hadoop-lzo. Notez qu'étant donné qu'il s'agit d'une bibliothèque tierce, Amazon EMR n'offre pas de support Developer sur la façon d'utiliser cet outil. Pour plus d'informations sur l'utilisation, consultez le fichier readme hadoop-lzo.

Import DynamoDB Data into Hive

L'implémentation de Hive fournie par Amazon EMR comprend des fonctionnalités que vous pouvez utiliser pour importer et exporter des données entre DynamoDB et un cluster Amazon EMR. Elles sont utiles si vos données d'entrée sont stockées dans DynamoDB. Pour de plus amples informations, veuillez consulter Exportation, importation, interrogation et jointure de tables dans DynamoDB à l'aide d'Amazon EMR.

Connect to Data with AWS DirectConnect

AWS Direct Connect est un service, que vous pouvez utiliser pour établir une connexion réseau dédiée privée à AWS à partir de votre centre de données, votre bureau ou votre environnement de colocation. Si vous avez de grandes quantités de données d'entrée, l'utilisation d'AWS Direct Connect peut réduire vos coûts de réseau, augmenter le débit de la bande passante et offrir une expérience réseau plus cohérente que les connexions basées sur Internet. Pour de plus amples informations, veuillez consulter AWS Direct Connect Guide de l'utilisateur.

Upload Large Amounts of Data with AWS Import/Export

AWS Import/Export est un service que vous pouvez utiliser pour transférer de grandes quantités de données à partir de périphériques de stockage physique dans AWS. Vous envoyez vos périphériques de stockage portables à AWS et AWS Import/Export transfère les données directement hors de vos périphériques de stockage à l'aide du réseau interne à haut débit d'Amazon. Votre charge de travail commence généralement le jour ouvrable suivant la réception de votre périphérique de stockage par AWS. Une fois l'importation ou l'exportation des données terminée, nous vous renvoyons votre périphérique de stockage. Pour des ensembles de données volumineux, le transfert des données AWS peut se révéler beaucoup plus rapide qu'un transfert via Internet, et plus économique que de mettre à niveau votre connectivité. Pour de plus amples informations, veuillez consulter AWS Import/Export Developer Guide.

Configure an Output Location

Le format de sortie le plus courant d'un cluster Amazon EMR est sous forme de fichiers texte, compressés ou non. En général, ceux-ci sont écrits dans un compartiment Amazon S3. Ce compartiment doit avoir été créé avant le lancement du cluster. Vous spécifiez le compartiment S3 comme emplacement de sortie lorsque vous lancez le cluster.

Pour de plus amples informations, consultez les rubriques suivantes :

Rubriques

- Create and Configure an Amazon S3 Bucket (p. 63)
- What formats can Amazon EMR return? (p. 65)
- Comment écrire des données dans un compartiment Amazon S3, qui ne vous appartient pas (p. 65)
- Compress the Output of your Cluster (p. 67)

Create and Configure an Amazon S3 Bucket

Amazon EMR (Amazon EMR) utilise Amazon S3 pour stocker les données d'entrée, les fichiers journaux et les données de sortie. Amazon S3 appelle ces emplacements de stockage compartiments. Les compartiments sont soumis à certaines restrictions et limitations pour se conformer aux exigences Amazon S3 et DNS. Pour de plus amples informations, consultez Limites et restrictions applicables aux compartiments dans le Guide du développeur Amazon Simple Storage Service.

Cette section vous montre comment utiliser Amazon S3AWS Management Console pour créer et définir des autorisations pour un compartiment Amazon S3. Cependant, vous pouvez également créer et définir

des autorisations pour un compartiment Amazon S3 à l'aide de l'API Amazon S3 ou de l'outil de ligne de commande Curl tiers. Pour de plus en amples informations sur l'utilisation de l'API Amazon S3 pour créer et configurer un compartiment Amazon S3, veuillez consulter Amazon Simple Storage Service API Reference.

Pour créer un compartiment Amazon S3 à l'aide de la console

1. Connectez-vous à AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Créer un compartiment.

   La boîte de dialogue Créer un compartiment s'ouvre.
3. Entrez un nom de compartiment, tel que ***DOC-EXAMPLE-BUCKET1***.

   Ce nom doit être globalement unique et il ne peut être similaire à un nom utilisé par un autre compartiment.
4. Sélectionnez la Région pour votre compartiment. Pour éviter de payer des frais de bande passante entre régions, créez le compartiment Amazon S3 dans la même région que votre cluster.

   Reportez-vous à Choose an AWS Region (p. 50) pour obtenir des conseils sur le choix d'une région.
5. Sélectionnez Create.

Vous avec créé un compartiment avec l'URI **s3n://*DOC-EXAMPLE-BUCKET1*/**.

> **Note**
>
> Si vous activez la journalisation dans l'assistant de création de compartiment, celui-ci active uniquement les journaux d'accès de compartiment, pas les journaux de cluster.

> **Note**
>
> Pour de plus amples informations sur la spécification de compartiments propres à une région, veuillez consulter Compartiments et régions dans le Guide du développeur Amazon Simple Storage Service et Points de terminaison de région disponibles pour les kits de développement logiciel AWS.

Une fois que vous avez créé le compartiment, vous pouvez définir les autorisations appropriées afférentes. Habituellement, vous (le propriétaire) vous donnez accès à la lecture et à l'écriture et l'accès à la lecture aux utilisateurs authentifiés.

Pour définir des autorisations sur un compartiment Amazon S3 à l'aide de la console

1. Connectez-vous à AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet Compartiment, ouvrez (à l'aide d'un clic droit) le compartiment que vous venez de créer.
3. Sélectionnez Propriétés.
4. Dans le panneau Propriétés, sélectionnez l'onglet Autorisations.
5. Choisissez Add more permissions (Ajouter plus d'autorisations).
6. Sélectionnez Authenticated Users (Utilisateurs authentifiés) dans le champ Grantee (Bénéficiaire).
7. A droite de la liste déroulante Grantee (Bénéficiaire), sélectionnez List (Liste).
8. Choisissez Save.

Vous avez créé un compartiment et limité les autorisations aux utilisateurs authentifiés.

Les compartiments Amazon S3 requis doivent avoir été créés pour que vous puissiez créer un cluster. Vous devez charger les scripts obligatoires ou les données référencées dans le cluster vers Amazon S3. Le tableau suivant décrit des exemples de données, de scripts et d'emplacements de fichier journal.

Informations	Exemple d'emplacement sur Amazon S3
script ou programme	`s3://DOC-EXAMPLE-BUCKET1/script/MapperScript.py`
fichiers journaux	`s3://DOC-EXAMPLE-BUCKET1/logs`
données d'entrée	`s3://DOC-EXAMPLE-BUCKET1/input`
données de sortie	`s3://DOC-EXAMPLE-BUCKET1/output`

What formats can Amazon EMR return?

Le format de sortie par défaut pour un cluster est du texte avec des paires de valeurs clés, écrites dans les lignes individuelles des fichiers texte. Il s'agit du format de sortie le plus couramment utilisé.

Si vos données de sortie doivent être écrites dans un format autre que les fichiers de texte par défaut, vous pouvez utiliser l'interface Hadoop `OutputFormat` pour spécifier d'autres types de sortie. Vous pouvez même créer une sous-classe de la classe `FileOutputFormat` pour gérer les types de données personnalisés. Pour plus d'informations, consultez http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/OutputFormat.html.

Si vous lancez un cluster Hive, vous pouvez utiliser un sérialiseur/désérialiseur (SerDe) pour générer des données à partir de HDFS dans un format donné. Pour plus d'informations, consultez https://cwiki.apache.org/confluence/display/Hive/SerDe.

Comment écrire des données dans un compartiment Amazon S3, qui ne vous appartient pas

Lorsque vous écrivez un fichier dans un compartiment Amazon Simple Storage Service (Amazon S3), vous êtes par défaut la seule personne capable de lire ce fichier. Nous supposons que vous allez écrire les fichiers dans vos propres compartiments. Ce paramètre par défaut protège la confidentialité de vos fichiers.

Toutefois, si vous exécutez un cluster et souhaitez que les données de sortie soient écrites dans le compartiment Amazon S3 d'un autre utilisateur AWS, et si vous voulez que l'autre utilisateur AWS soit capable de lire ces données, deux actions doivent être mises en œuvre :

- L'autre utilisateur AWS doit vous accorder les autorisations de lecture pour son compartiment Amazon S3. Le cluster que vous lancez utilise vos informations d'identification AWS. Ainsi, tous les clusters que vous lancez seront également en mesure d'écrire dans le compartiment de cet autre utilisateur AWS.
- Vous devez définir les autorisations en lecture pour l'autre utilisateur AWS sur les fichiers que le cluster ou vous-même écrivez dans le compartiment Amazon S3. La manière la plus simple de définir ces autorisations en lecture consiste à utiliser des listes de contrôle d'accès prédéfinies, ensemble de stratégies d'accès prédéfinies défini par Amazon S3.

Pour plus d'informations sur la façon dont l'autre utilisateur AWS peut vous accorder des autorisations pour écrire des fichiers dans son compartiment Amazon S3, consultez Gérer les autorisations d'un compartiment dans le Amazon Simple Storage Service Guide de l'utilisateur de la console.

Pour que votre cluster utilise des listes de contrôle d'accès prédéfinies lorsqu'il écrit des fichiers dans Amazon S3, définissez l'option de configuration du cluster `fs.s3.canned.acl` sur la liste de contrôle

d'accès à utiliser. Le tableau suivant répertorie les listes de contrôle d'accès prédéfinies actuellement définies.

Liste ACL prête à l'emploi	Description :
`AuthenticatedRead`	Spécifie que le propriétaire reçoit l'autorisation `Permission.FullControl` et que le bénéficiaire du groupe reçoit l'autorisation `GroupGrantee.AuthenticatedUsers` `Permission.Read`.
`BucketOwnerFullControl`	Spécifie que le propriétaire du seau est accordé `Permission.FullControl`. Le propriétaire du seau n'est pas nécessairement le même que le propriétaire de l'objet.
`BucketOwnerRead`	Spécifie que le propriétaire du seau est accordé `Permission.Read`. Le propriétaire du seau n'est pas nécessairement le même que le propriétaire de l'objet.
`LogDeliveryWrite`	Spécifie que le propriétaire reçoit l'autorisation `Permission.FullControl` et que le bénéficiaire du groupe `GroupGrantee.LogDelivery` reçoit l'autorisation `Permission.Write`, afin que les journaux d'accès puissent être transmis.
`Private`	Spécifie que le propriétaire reçoit l'autorisation `Permission.FullControl`.
`PublicRead`	Spécifie que le propriétaire reçoit l'autorisation `Permission.FullControl` et que le bénéficiaire du groupe reçoit l'autorisation `GroupGrantee.AllUsers` `Permission.Read`.
`PublicReadWrite`	Spécifie que le propriétaire reçoit l'autorisation `Permission.FullControl` et que le bénéficiaire du groupe `GroupGrantee.AllUsers` reçoit les autorisations `Permission.Read` et `Permission.Write`.

Il existe différentes façons de définir les options de configuration du cluster, en fonction du type de cluster que vous exécutez. Les procédures suivantes montrent comment définir les options pour les cas les plus courants.

Pour écrire des fichiers à l'aide de listes ACL prédéfinies dans Hive

- À l'invite de commande Hive, définissez l'option de configuration `fs.s3.canned.acl` sur la liste ACL prédéfinie souhaitée afin que le cluster soit défini sur les fichiers écrits dans Amazon S3. Pour accéder à l'invite de commande Hive, connectez-vous au nœud maître à l'aide de SSH, puis tapez Hive à l'invite de commande Hadoop. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368),

  L'exemple suivant définit l'option de configuration `fs.s3.canned.acl` sur `BucketOwnerFullControl`, ce qui accorde au propriétaire du compartiment Amazon S3 un contrôle complet sur le fichier. Notez que la commande définie est sensible à la casse et ne contient pas de guillemet ni d'espace.

```
hive> set fs.s3.canned.acl=BucketOwnerFullControl;
create table acl (n int) location 's3://acltestbucket/acl/';
insert overwrite table acl select count(*) from acl;
```

Les deux dernières lignes de l'exemple créent une table qui est stockée dans Amazon S3 et écrivent des données dans la table.

Pour écrire des fichiers à l'aide de listes ACL prédéfinies dans Pig

- A l'invite de commande Pig, définissez l'option de configuration `fs.s3.canned.acl` sur la liste ACL prédéfinie souhaitée afin que le cluster soit défini sur les fichiers écrits dans Amazon S3. Pour accéder à l'invite de commande Pig, connectez-vous au nœud maître à l'aide de SSH, puis tapez Pig à l'invite de commande Hadoop. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368),

  L'exemple suivant définit l'option de configuration `fs.s3.canned.acl` sur BucketOwnerFullControl, ce qui accorde au propriétaire du compartiment Amazon S3 un contrôle complet sur le fichier. Notez que la commande définie comprend un espace avant le nom de la liste ACL prédéfinie et ne contient aucun guillemet.

```
pig> set fs.s3.canned.acl BucketOwnerFullControl;
store some data into 's3://acltestbucket/pig/acl';
```

Pour écrire des fichiers à l'aide de listes ACL prédéfinies dans un fichier JAR personnalisé

- Définissez l'option de configuration `fs.s3.canned.acl` à l'aide de Hadoop avec l'indicateur -D. Cela est illustré dans l'exemple suivant.

```
hadoop jar hadoop-examples.jar wordcount
-Dfs.s3.canned.acl=BucketOwnerFullControl s3://mybucket/input s3://mybucket/output
```

Compress the Output of your Cluster

Rubriques

- Output Data Compression (p. 67)
- Intermediate Data Compression (p. 68)
- Using the Snappy Library with Amazon EMR (p. 68)

Output Data Compression

Cela compresse les données de sortie de votre tâche Hadoop. Si vous utilisez TextOutputFormat, le résultat est un fichier texte au format gzip. Si vous écrivez à SequenceFiles, le résultat est alors un SequenceFile qui est compressé en interne. Cela peut être activé en définissant le paramètre de configuration mapred.output.compress sur true.

Si vous exécutez une tâche de diffusion en continu, vous pouvez l'activer en transmettant ces arguments à la tâche de diffusion en continu.

```
-jobconf mapred.output.compress=true
```

Vous pouvez également utiliser une action d'amorçage pour compresser automatiquement toutes les sorties de la tâche. Voici comment procéder avec le client Ruby.

```
--bootstrap-actions s3://elasticmapreduce/bootstrap-actions/configure-hadoop \
--args "-s,mapred.output.compress=true"
```

Enfin, en cas d'écriture d'un JAR personnalisé, vous pouvez activer la compression de sortie avec la ligne suivante lors de la création de votre tâche.

```
FileOutputFormat.setCompressOutput(conf, true);
```

Intermediate Data Compression

Si votre tâche lit de façon aléatoire un volume important de données des mappeurs vers les réducteurs, vous pouvez voir une amélioration des performances en activant la compression intermédiaire. Compressez la sortie de la carte et décompressez-la quand elle arrive sur le nœud de noyau. Le paramètre de configuration est mapred.compress.map.output. Vous pouvez activer cela de la même manière que la compression de sortie.

Lorsque vous écrivez un JAR personnalisé, utilisez la commande suivante :

```
conf.setCompressMapOutput(true);
```

Using the Snappy Library with Amazon EMR

Snappy est une bibliothèque de compression et de décompression qui est optimisée pour la vitesse. Elle est disponible sur les AMI Amazon EMR version 2.0 et versions ultérieures et est utilisée en tant que valeur par défaut pour la compression intermédiaire. Pour plus d'informations sur Snappy, consultez http://code.google.com/p/snappy/.

Planification et configuration des nœuds maîtres

Lorsque vous lancez un cluster EMR, vous pouvez choisir d'avoir un ou trois nœuds maîtres dans votre cluster. Le lancement d'un cluster avec trois nœuds maîtres est uniquement pris en charge par Amazon EMR version 5.23.0 et versions ultérieures.

Un Cluster EMR doté de plusieurs nœuds principaux fournit les avantages clés suivants :

- Le nœud maître n'est plus un point de défaillance unique. Si l'un des nœuds maîtres échoue, le cluster utilise les deux autres nœuds maîtres et s'exécute sans interruption. En attendant, Amazon EMR remplace automatiquement le nœud maître en échec par un nouveau qui est configuré avec les mêmes configurations et actions d'amorçage.
- EMR autorise les fonctionnalités de haute disponibilité Hadoop de HDFS NameNode et du gestionnaire de ressources YARN et prend en charge la haute disponibilité pour d'autres applications open source.

  Pour plus d'informations sur la façon dont un Cluster EMR doté de plusieurs nœuds principaux prend en charge les applications open source et d'autres fonctions EMR, consultez Applications et fonctionnalités prises en charge (p. 69).

> **Note**
>
> Le cluster peut résider uniquement dans une zone de disponibilité ou un sous-réseau.

Cette section fournit des informations sur les applications et les fonctions d'un Cluster EMR doté de plusieurs nœuds principaux prises en charge, ainsi que les détails de configuration, les bonnes pratiques et les aspects à prendre en compte lors du lancement du cluster.

Rubriques

- Applications et fonctionnalités prises en charge (p. 69)
- Lancement d'un cluster EMR avec plusieurs nœuds maîtres (p. 74)
- Considérations et bonnes pratiques (p. 76)

Applications et fonctionnalités prises en charge

Cette rubrique fournit des informations sur les fonctionnalités de haute disponibilité Hadoop de HDFS NameNode et du gestionnaire de ressources YARN dans un cluster EMR et comment les fonctions de haute disponibilité fonctionnent avec les applications open source et d'autres fonctionnalités EMR.

Haute disponibilité de HDFS

Un Cluster EMR doté de plusieurs nœuds principaux permet la fonctionnalité de haute disponibilité HDFS NameNode dans Hadoop. Pour de plus amples informations, veuillez consulter Haute disponibilité HDFS.

Dans un cluster EMR, NameNode s'exécute uniquement sur deux des trois nœuds maîtres. Un NameNode est dans l'état `active` et l'autre est dans l'état `standby`. Si le nœud maître avec le NameNode `active` échoue, EMR démarre un processus de basculement automatique HDFS. Le nœud maître avec le NameNode `standby` devient `active` et prend en charge toutes les opérations de client dans le cluster. EMR remplace le nœud maître en échec par un nouveau, puis le rejoint en tant que `standby`.

Si vous avez besoin de savoir quel NameNode est `active`, vous pouvez utiliser SSH pour vous connecter à un nœud maître dans le cluster et exécuter la commande suivante :

```
hdfs haadmin -getAllServiceState
```

La sortie répertorie les deux nœuds où NameNode est installé et leur état. Par exemple,

```
ip-##-#-#-##1.ec2.internal:8020 active
ip-##-#-#-##2.ec2.internal:8020 standby
```

Haute disponibilité du gestionnaire de ressources YARN

Un Cluster EMR doté de plusieurs nœuds principaux autorise la fonctionnalité de haute disponibilité du gestionnaire de ressources YARN dans Hadoop. Pour de plus amples informations, veuillez consulter Haute disponibilité du gestionnaire de ressource.

Dans un Cluster EMR doté de plusieurs nœuds principaux, le gestionnaire de ressources YARN s'exécute sur les trois nœuds maîtres. Un gestionnaire de ressources se trouve dans l'état `active`, et les deux autres sont dans l'état `standby`. Si le nœud maître avec le gestionnaire de ressources `active` échoue, EMR démarre un processus de basculement automatique. Un nœud maître avec un gestionnaire de ressources `standby` prend en charge toutes les opérations. EMR remplace le nœud maître en échec par un nouveau, puis rejoint le quorum du gestionnaire de ressources en tant que `standby`.

Vous pouvez vous connecter à "http ://*`master-public-dns-name`*:8088/cluster » pour n'importe quel nœud maître, qui vous dirige automatiquement vers le `active` du gestionnaire de ressources. Pour déterminer quel gestionnaire de ressources est `active`, utilisez SSH pour vous connecter à un nœud

maître dans le cluster. Ensuite, exécutez la commande suivante pour obtenir la liste des trois nœuds maîtres et leur statut :

```
yarn rmadmin -getAllServiceState
```

Applications prises en charge dans un cluster EMR avec plusieurs nœuds maîtres

Vous pouvez installer et exécuter les applications suivantes sur un Cluster EMR doté de plusieurs nœuds principaux. Pour chaque application, le processus de basculement du nœud principal (ou nœud maître) varie.

Application	Disponibilité lors du basculement du nœud maître	Remarques:
Flink	Disponibilité non affectée par le basculement du nœud maître	Les tâches Flink sur Amazon EMR s'exécutent en tant qu'applications YARN. Les gestionnaires de tâches de Flink s'exécutent en tant que maîtres d'application (ApplicationMasters) de YARN sur les nœuds principaux. Le gestionnaires de tâches n'est pas affecté par le processus de basculement du nœud principal. Si vous utilisez Amazon EMR version 5.27.0 ou antérieure, le gestionnaire de tâches constitue un point de défaillance unique. Lorsque le gestionnaire de tâches échoue, il perd tous les états de tâche et ne reprend pas les tâches en cours d'exécution. Vous pouvez activer la haute disponibilité du gestionnaire de tâches en configurant le nombre de tentatives de l'application et les points de contrôle, et en activant ZooKeeper comme stockage d'état pour Flink. Pour plus d'informations, consultez Configuration de Flink sur un cluster EMR doté de plusieurs nœuds principaux. À partir d'Amazon EMR version 5.28.0, aucune configuration manuelle n'est nécessaire pour activer la haute disponibilité du gestionnaire de tâches.
Ganglia	Disponibilité non affectée par le basculement du nœud maître	Ganglia est disponible sur tous les nœuds maîtres. Ainsi, Ganglia peut continuer à s’exécuter pendant le processus de basculement du nœud maître.
Hadoop	Haute disponibilité	HDFS NameNode et le gestionnaire de ressources YARN basculent automatiquement vers le nœud en état de veille lorsque le nœud maître actif échoue.
HBase	Haute disponibilité	HBase bascule automatiquement vers le nœud en veille lorsque le nœud maître actif échoue. Si vous vous connectez à HBase via un serveur Thrift ou REST, vous devez passer à un autre nœud maître lorsque le nœud maître actif échoue.

Application	Disponibilité lors du basculement du nœud maître	Remarques:
HCatalog	Disponibilité non affectée par le basculement du nœud maître	HCatalog est construit sur le métastore Hive, qui existe en dehors du cluster. HCatalog reste disponible pendant le processus de basculement du nœud maître.
JupyterHub	Haute disponibilité	JupyterHub est installé sur les trois instances principales. Il est fortement recommandé de configurer la persistance du bloc-notes pour éviter la perte du bloc-notes en cas de défaillance du nœud principal. Pour plus d'informations, consultez Configuration de persistance pour les blocs-notes dans Amazon S3.
Livy	Haute disponibilité	Livy est installé sur les trois nœuds maîtres. Lorsque le nœud maître actif échoue, vous perdez l'accès à la session Livy actuelle et vous devez créer une nouvelle session Livy sur un autre nœud maître ou sur le nouveau nœud de remplacement.
Mahout	Disponibilité non affectée par le basculement du nœud maître	Étant donné que Mahout n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud maître.
MXNet	Disponibilité non affectée par le basculement du nœud maître	Étant donné que MXNet n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud principal.
Phoenix	Haute disponibilité	QueryServer de Phoenix s'exécute uniquement sur l'un des trois nœuds principaux (maître). Phoenix sur les trois maîtres est configuré pour connecter le serveur Phoenix QueryServer. Vous pouvez trouver l'adresse IP privée du serveur de requête de Phoenix en utilisant le fichier `/etc/phoenix/conf/phoenix-env.sh`
Pig	Disponibilité non affectée par le basculement du nœud maître	Étant donné que Pig n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud maître.
Spark	Haute disponibilité	Toutes les applications Spark s’exécutent dans des conteneurs YARN et peuvent réagir au basculement du nœud maître de la même manière que les fonctionnalités de haute disponibilité de YARN.
Sqoop	Haute disponibilité	Par défaut, sqoop-job et sqoop-metastore stockent les données (descriptions de tâche) sur le disque local du maître qui exécute la commande. Si vous souhaitez enregistrer des données de metastore dans une base de données externe, consultez la documentation Apache Sqoop
Tez	Haute disponibilité	Puisque les conteneurs Tez s'exécutent sur YARN, Tez se comporte de la même manière que YARN lors du processus de basculement du nœud maître.

Application	Disponibilité lors du basculement du nœud maître	Remarques:
TensorFlow	Disponibilité non affectée par le basculement du nœud maître	Étant donné que TensorFlow n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud principal.
Zeppelin	Haute disponibilité	Zeppelin est installé sur les trois nœuds principaux. Zeppelin stocke les notes et les configurations d'interpréteur dans HDFS par défaut pour éviter la perte de données. Les sessions d'interpréteur sont complètement isolées sur les trois instances principales (maître). Les données de session seront perdues en cas d'échec du maître. Il est recommandé de ne pas modifier simultanément la même note sur différentes instances principales.
ZooKeeper	Haute disponibilité	ZooKeeper est la base de la fonctionnalité de basculement automatique de HDFS. ZooKeeper fournit un service hautement disponible pour maintenir les données de coordination, avertir les clients des modifications apportées à ces données, et surveiller des clients pour détecter les pannes. Pour de plus amples informations, veuillez consulter Basculement automatique de HDFS.

Pour exécuter les applications suivantes dans un cluster EMR doté de plusieurs nœuds principaux, vous devez configurer une base de données externe. La base de données externe existe en dehors du cluster et rend les données persistantes pendant le processus de basculement du nœud principal. Pour les applications suivantes, les composants de service se rétabliront automatiquement pendant le processus de basculement du nœud principal, mais les tâches actives peuvent échouer et doivent être relancées.

Application	Disponibilité lors du basculement du nœud maître	Remarques:
Hive	Haute disponibilité pour les composants de service uniquement	Un metastore externe pour Hive est requis. Pour de plus amples informations, veuillez consulter Configuration d'un métastore externe pour Hive.
Hue	Haute disponibilité pour les composants de service uniquement	Une base de données externe pour Hue est requise. Pour plus d'informations, consultez Utilisation de Hue avec une base de données distante dans Amazon RDS.
Oozie	Haute disponibilité pour les composants de service uniquement	Une base de données externe pour Oozie est requise. Pour plus d'informations, consultez Utilisation d'Oozie avec une base de données distante dans Amazon RDS. Le serveur oozie-server est installé sur un seul nœud principal. Par contre, le client oozie-client est installé sur les trois nœuds principaux. Les clients oozie-sont configurés pour se connecter au serveur oozie-server correct par défaut. Vous pouvez trouver le nom DNS privé du nœud principal sur lequel le serveur oozie-server est installé en vérifiant la variable `OOZIE_URL` sur n'importe

Application	Disponibilité lors du basculement du nœud maître	Remarques:
		quel nœud principal dans le fichier `/etc/oozie/conf.dist/oozie-client-env.sh`.
PrestoDB ou PrestoSQL	Haute disponibilité pour les composants de service uniquement	Un métastore Hive externe pour PrestoDB (ou pour PrestoSQL sur emr-6.1.0 et versions ultérieures) est requis. Vous pouvez utiliser Presto avec le AWS Glue Data Catalog ou utiliser une base de données MySQL externe pour Hive.

Note

Lorsqu'un nœud maître échoue, votre Java Database Connectivity (JDBC) ou Open Database Connectivity (ODBC) met fin à sa connexion au nœud maître. Vous pouvez vous connecter à l'un des autres nœuds maîtres pour continuer votre travail puisque le démon métastore Hive s'exécute sur tous les nœuds maîtres. Ou vous pouvez attendre le remplacement du nœud maître en échec.

Comment les fonctionnalités EMR fonctionnent dans un cluster avec plusieurs nœuds maîtres

Connexion aux nœuds maîtres à l'aide de SSH

Vous pouvez vous connecter à l'un des trois nœuds maîtres dans un cluster EMR à l'aide de SSH de la même manière que vous vous connectez à un seul nœud maître. Pour plus d'informations, consultez Se connecter au nœud principal à l'aide de SSH.

Si un nœud maître échoue, votre connexion SSH au nœud maître se termine. Pour continuer votre travail, vous pouvez vous connecter à l'un des deux autres nœuds maîtres. Vous pouvez également accéder au nouveau nœud maître une fois qu'EMR remplace celui en échec par un nouveau.

Note

L'adresse IP privée pour le nœud maître de remplacement reste la même que la précédente. L'adresse IP publique pour le nœud maître de remplacement peut changer. Vous pouvez récupérer les nouvelles adresses IP dans la console ou à l'aide de la commande `describe-cluster` de l'interface de ligne de commande AWS.
NameNode s'exécute uniquement sur deux des nœuds maîtres. Cependant, vous pouvez exécuter les commandes de l'interface de ligne de commande `hdfs` et exploiter des tâches pour accéder à HDFS sur les trois nœuds maîtres.

Utilisation des étapes dans un cluster EMR avec plusieurs nœuds maîtres

Vous pouvez soumettre des étapes à un Cluster EMR doté de plusieurs nœuds principaux de la même manière que vous utilisez des étapes dans un cluster avec un seul nœud maître. Pour de plus amples informations, veuillez consulter Soumission de travail à un cluster.

Voici des considérations relatives à l'utilisation des étapes dans un Cluster EMR doté de plusieurs nœuds principaux :

- Si un nœud maître échoue, les étapes en cours d'exécution sur le nœud maître sont marquées comme FAILED. Toutes les données écrites en local sont perdues. Toutefois, le statut FAILED peut ne pas refléter l'état réel des étapes.
- Si une étape en cours d'exécution a démarré une application YARN lorsque le nœud maître échoue, l'étape peut continuer et réussir grâce au basculement automatique du nœud maître.

- Il est recommandé de vérifier le statut des étapes en faisant référence à la sortie des tâches. Par exemple, les tâches MapReduce utilisent un fichier `_SUCCESS` pour déterminer si la tâche est terminée avec succès.
- Il est recommandé de définir le paramètre ActionOnFailure sur CONTINUE ou CANCEL_AND_WAIT, au lieu de TERMINATE_JOB_FLOW ou TERMINATE_CLUSTER.

Fonctions non prises en charge dans un cluster EMR avec plusieurs nœuds maîtres

Les fonctionnalités EMR suivantes ne sont actuellement pas disponibles dans un Cluster EMR doté de plusieurs nœuds principaux :

- Blocs-notes EMR
- Parcs d'instances
- Accès en un clic au serveur d'historique Spark permanent

> **Note**
>
> Pour utiliser l'authentification Kerberos dans votre cluster, vous devez configurer un KDC externe. À partir de Amazon EMR version 5.27.0, vous pouvez configurer le chiffrement transparent HDFS sur un Cluster EMR doté de plusieurs nœuds principaux. Pour plus d'informations, consultez Chiffrement transparent dans HDFS sur Amazon EMR.

Lancement d'un cluster EMR avec plusieurs nœuds maîtres

Cette rubrique fournit des détails de configuration et des exemples de lancement d'un Cluster EMR doté de plusieurs nœuds principaux.

Prerequisites

- Vous pouvez lancer un Cluster EMR doté de plusieurs nœuds principaux dans des sous-réseaux VPC publics et privés. EC2-Classic n'est pas pris en charge. Pour lancer un Cluster EMR doté de plusieurs nœuds principaux dans un sous-réseau public, vous devez activer les instances dans ce sous-réseau pour recevoir une adresse IP publique en sélectionnant Auto-assign IPv4 dans la console ou en exécutant la commande suivante. Remplacez *22XXXX01* avec votre ID de sous-réseau.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-22XXXX01 --map-public-ip-on-launch
```

- Pour exécuter Hive, Hue ou Oozie sur un Cluster EMR doté de plusieurs nœuds principaux, vous devez créer un metastore externe pour Hive. Pour plus d'informations, consultez Configuration d'une metastore externe pour Hive, Utilisation de Hue avec une base de données distante dans Amazon RDS ou Apache Oozie.
- Pour utiliser l'authentification Kerberos dans votre cluster, vous devez configurer un KDC externe. Pour de plus amples informations, veuillez consulter Configuration de Kerberos sur Amazon EMR.

Lancer un cluster EMR doté de plusieurs nœuds maîtres

Vous devez spécifier une valeur de nombre d'instances de trois pour le groupe d'instance du nœud maître lorsque vous lancez un Cluster EMR doté de plusieurs nœuds principaux. Les exemples suivants montrent comment lancer le cluster à l'aide de l'AMI par défaut ou une AMI personnalisée.

Note

Vous devez spécifier l'ID de sous-réseau lorsque vous lancez un Cluster EMR doté de plusieurs nœuds principaux à l'aide de l'AWS CLI. Remplacez *22XXXX01* avec votre ID de sous-réseau dans les exemples suivants.

Example –Lancement d'un Cluster EMR doté de plusieurs nœuds principaux à l'aide d'une AMI par défaut

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-5.30.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

Example –Lancement d'un Cluster EMR doté de plusieurs nœuds principaux à l'aide d'une AMI personnalisée

```
aws emr create-cluster \
--name "custom-ami-ha-cluster" \
--release-label emr-5.30.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID
```

Example –Lancement d'un Cluster EMR doté de plusieurs nœuds principaux avec un métastore Hive externe

Pour exécuter Hive sur un Cluster EMR doté de plusieurs nœuds principaux, vous devez spécifier un métastore externe pour Hive, comme l'exemple suivant le montre,

1. Créez un fichier hiveConfiguration.json temporaire qui contient les informations d'identification pour votre métastore Hive.

```
[
  {
    "Classification": "hive-site",
    "Properties": {
      "javax.jdo.option.ConnectionURL": "jdbc:mysql:\/\/hostname:3306\/hive?
createDatabaseIfNotExist=true",
      "javax.jdo.option.ConnectionDriverName": "org.mariadb.jdbc.Driver",
      "javax.jdo.option.ConnectionUserName": "username",
      "javax.jdo.option.ConnectionPassword": "password"
    }
  }
]
```

2. Lancez le cluster avec le métastore Hive.

```
aws emr create-cluster \
--name "ha-cluster-with-hive-metastore" \
--release-label emr-5.30.1 \
```

```
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name= Spark Name=Hive \
--configurations ./hiveConfiguration.json
```

Arrêter un cluster EMR avec plusieurs nœuds maîtres

Pour arrêter un Cluster EMR doté de plusieurs nœuds principaux, vous devez désactiver la protection d'arrêt avant d'arrêter le cluster, comme l'exemple suivant le montre. Remplacez *j-3KVTXXXXXX7UG* avec votre ID de cluster.

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
aws emr terminate-clusters --cluster-id j-3KVTXXXXXX7UG
```

Considérations et bonnes pratiques

Limitations d'un Cluster EMR doté de plusieurs nœuds principaux

- Si les deux nœuds maîtres échouent simultanément, EMR ne peut pas récupérer le cluster.
- Les clusters EMR avec plusieurs nœuds maîtres ne sont pas tolérants aux défaillances des zones de disponibilité. Dans le cas d'une panne d'une zone de disponibilité, vous perdez l'accès au cluster EMR.
- EMR ne garantit pas les fonctionnalités de haute disponibilité des d'applications open source autres que celles spécifiées dans Applications prises en charge dans un cluster EMR avec plusieurs nœuds maîtres (p. 70).

Considérations relatives à la configuration d'un sous-réseau :

- Un Cluster EMR doté de plusieurs nœuds principaux peut résider uniquement dans une zone de disponibilité ou un sous-réseau. EMR ne peut pas remplacer un nœud maître en échec si le sous-réseau est entièrement utilisé ou congestionné dans le cas d'un basculement. Pour éviter ce scénario, il est recommandé de dédier l'ensemble d'un sous-réseau à un cluster Amazon EMR. En outre, assurez-vous qu'il y ait suffisamment d'adresses IP privées disponibles dans le sous-réseau.

Considérations relatives à la configuration de nœuds principaux :

- Pour vous assurer que le groupe d'instances du nœud principal dispose également d'une haute disponibilité, il est recommandé de lancer au moins quatre nœuds principaux. Si vous décidez de lancer un plus petit cluster avec trois nœuds maîtres ou moins, configurez HDFS avec suffisamment de réplication DFS en définissant `dfs.replication parameter` sur au moins 2. Pour de plus amples informations, veuillez consulter Configuration de HDFS.

Considérations relatives à la configuration d'alarmes sur les métriques :

- EMR ne fournit pas actuellement les métriques spécifiques à une application donnée sur HDFS ou YARN. Il est recommandé de configurer des alarmes pour surveiller le nœud maître d'instances. Vous pouvez configurer les alarmes à l'aide des métriques CloudWatch suivantes : `MultiMasterInstanceGroupNodesRunning`, `MultiMasterInstanceGroupNodesRunningPercentage`ou `MultiMasterInstanceGroupNodesRequested`. Vous serez averti en cas de défaillance du nœud principal et de remplacement de. Par exemple,

- Si le `MultiMasterInstanceGroupNodesRunningPercentage` est inférieur à 1,0 et supérieur à 0,5, le cluster peut avoir perdu un nœud maître. Dans ce cas, EMR tente de remplacer un nœud maître.
- Si le `MultiMasterInstanceGroupNodesRunningPercentage` passe en dessous de 0,5, deux nœuds maîtres peuvent avoir échoué. Dans ce cas, le quorum est perdu et le cluster ne peut pas être récupéré. Une intervention manuelle est requise pour migrer des données en dehors de ce cluster.

Pour de plus amples informations, veuillez consulter Configuration d'alarmes sur les métriques.

Clusters EMR sur AWS Outposts

À partir d'Amazon EMR version 5.28.0, vous pouvez créer et exécuter des clusters EMR sur AWS Outposts. AWS Outposts active des services AWS natifs, une infrastructure et des modèles d'exploitation dans des installations sur site. Dans les environnements AWS Outposts, vous pouvez utiliser les mêmes API, outils et infrastructure AWS que ceux que vous utilisez dans le cloud AWS. Amazon EMR sur AWS Outposts est idéal pour les charges de travail à faible latence qui ont besoin d'être exécutées à proximité des données et des applications sur site. Pour de plus amples informations sur AWS Outposts, veuillez consulter le Guide de l'utilisateur AWS Outposts.

Prérequis

Les conditions suivantes sont requises pour utiliser Amazon EMR sur AWS Outposts :

- Vous devez avoir installé et configuré AWS Outposts dans votre centre de données sur site.
- Vous devez avoir une connexion réseau fiable entre votre environnement Outpost et une région AWS.
- Vous devez disposer d'une capacité suffisante pour les types d'instance pris en charge par EMR dans votre Outpost.

Limites

L'utilisation d'Amazon EMR sur AWS Outposts présente les limites suivantes :

- Les instances à la demande constituent la seule option prise en charge pour les instances Amazon EC2. Les instances Spot ne sont pas disponibles pour Amazon EMR sur AWS Outposts.
- Si vous avez besoin de volumes de stockage Amazon EBS supplémentaires, seul le volume à usage général SSD (GP2) est pris en charge.
- Seuls les types d'instance suivants sont pris en charge par Amazon EMR sur AWS Outposts:

Classe d'instance	Types d'instance
Usage général	m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge
Optimisé pour le calcul	c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge| c5d.9xlarge | c5d.18xlarge
Optimisé pour la mémoire	r5.xlarge | r5.2xlarge | r5.4xlarge | r5.12xlarge | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge
Optimisé pour le stockage	i3en.xlarge | i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge | i3en.24xlarge

Considérations relatives à la connectivité réseau

- Si la connectivité réseau entre votre Outpost et sa région AWS est perdue, vos clusters continuent de s'exécuter. Toutefois, vous ne pouvez pas créer de nouveaux clusters ou effectuer de nouvelles actions sur les clusters existants tant que la connectivité n'est pas rétablie. En cas de défaillance d'une instance, celle-ci ne sera pas remplacée automatiquement. En outre, les actions comme l'ajout d'étapes à un cluster en cours d'exécution, la vérification du statut d'exécution de l'étape, ainsi que l'envoi de métriques et d'événements CloudWatch seront retardées.
- Nous vous recommandons de fournir une connectivité réseau fiable et hautement disponible entre votre Outpost et la région AWS. Si la connectivité réseau entre votre Outpost et sa région AWS est perdue pendant plus de quelques heures, les clusters pour lesquels la protection contre la résiliation est activée continueront à s'exécuter et les clusters pour lesquels la protection contre la résiliation est désactivée pourront être résiliés.
- Si la connectivité réseau sera affectée par une maintenance de routine, nous recommandons d'activer la protection contre la résiliation de manière proactive. Plus généralement, une interruption de connectivité signifie que toutes les dépendances externes qui ne sont pas locales au réseau Outpost ou client ne seront pas accessibles. Cela inclut Amazon S3, DynamoDB utilisé avec la vue de cohérence EMRFS et Amazon RDS si une instance dans la région est utilisée pour un Cluster EMR doté de plusieurs nœuds principaux.

Création d'un cluster Amazon EMR sur AWS Outposts

La création d'un cluster Amazon EMR sur AWS Outposts est similaire à la création d'un cluster Amazon EMR dans le cloud AWS. Lorsque vous créez un cluster Amazon EMR sur AWS Outposts, vous devez spécifier le sous-réseau Amazon EC2 qui associé à votre Outpost.

Un Amazon VPC peut couvrir toutes les zones de disponibilité d'une région AWS. Amazon VPC sont des extensions des zones de disponibilité, et vous pouvez étendre un AWS Outposts d'un compte pour couvrir plusieurs zones de disponibilité, ainsi que les emplacements Outpost associés. Lorsque vous configurez votre Outpost, vous lui associez un sous-réseau pour étendre votre environnement VPC régional à votre installation sur site. Les instances Outpost et les services associés apparaissent dans le cadre de votre VPC régional de façon similaire à une zone de disponibilité avec des sous-réseaux associés. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Outposts.

Console

Pour créer un cluster Amazon EMR sur AWS Outposts avec l'AWS Management Console, spécifiez un sous-réseau Amazon EC2 associé à votre Outpost.

1. Ouvrez la console Amazon EMR.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Sous Software Configuration (Configuration logicielle), pour Release (Version), choisissez 5.28.0 ou une version ultérieure.
5. Sous Hardware Configuration (Configuration matérielle), pour EC2 Subnet (Sous-réseau EC2), sélectionnez un sous-réseau EC2 avec un ID Outpost au format suivant : op-123456789.
6. Choisissez le type d'instance ou ajoutez des volumes de stockage Amazon EBS pour des groupes d'instances ou des parcs d'instances uniformes. Le volume Amazon EBS limité et les types d'instance sont pris en charge pour Amazon EMR sur AWS Outposts.

AWS CLI

Pour créer un cluster Amazon EMR sur AWS Outposts avec l'AWS CLI, spécifiez un sous-réseau EC2 associé à votre Outpost.

L'exemple suivant crée un cluster Amazon EMR sur un Outpost. Remplacez *subnet-22XXXX01* par un sous-réseau EC2 associé à votre Outpost.

```
aws emr create-cluster \
--name "Outpost cluster" \
--release-label emr-5.30.1 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey SubnetId=subnet-22XXXX01 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Clusters EMR dans les Zones locales AWS

À partir de la version 5.28.0 d'Amazon EMR, vous pouvez créer et exécuter des clusters Amazon EMR dans un sous-réseau de Zones locales AWS en tant qu'extension logique d'une région AWS prenant en charge les Zones locales. Une Zone locale permet de rapprocher les fonctionnalités Amazon EMR et un sous-ensemble de services AWS, tels que les services de calcul et de stockage, des utilisateurs afin de fournir un accès à très faible latence aux applications exécutées localement. Pour obtenir la liste des Zones locales disponibles, consultez Zones locales AWS. Pour plus d'informations sur l'accès aux Zones locales AWS disponibles, consultez Régions, zones de disponibilité et zones locales.

Types d'instance pris en charge

Les types d'instance suivants sont disponibles pour les clusters Amazon EMR dans les Zones locales. La disponibilité du type d'instance peut varier selon la région.

Classe d'instance	Types d'instance
Usage général	m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge
Optimisé pour le calcul	c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge| c5d.9xlarge | c5d.18xlarge
Optimisé pour la mémoire	r5.xlarge | r5.2xlarge | r5.4xlarge | r5.12xlarge | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge
Optimisé pour le stockage	i3en.xlarge | i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge | i3en.24xlarge

Création d'un cluster Amazon EMR dans des Zones locales

Pour créer un cluster Amazon EMR dans des Zones locales AWS, lancez le cluster Amazon EMR dans un sous-réseau Amazon VPC associé à une Zone locale. Vous pouvez accéder au cluster en utilisant le nom de la Zone locale, par exemple us-west-2-lax-1a dans la console USA Ouest (Oregon).

Les zones locales ne prennent actuellement pas en charge Amazon EMR Notebooks ni les connexions directement à Amazon EMR à l'aide du point de terminaison du VPC d'interface (AWS PrivateLink).

Pour créer un cluster EMR dans une Zone locale l'aide de la console Amazon EMR

Pour créer un cluster Amazon EMR dans des Zones locales avec AWS Management Console, spécifiez un sous-réseau Amazon EC2 associé à votre Zone locale.

1. Ouvrez la console Amazon EMR.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Sous Software Configuration (Configuration logicielle), pour Release (Version), choisissez 5.28.0 ou une version ultérieure.
5. Sous Hardware Configuration (Configuration matérielle), pour EC2 Subnet (Sous-réseau EC2), sélectionnez un sous-réseau EC2 avec un ID de Zone locale au format suivant : sous-réseau 123abc | us-west-2-lax-1a.
6. Ajoutez des volumes de stockage Amazon EBS pour des groupes d'instances ou des parcs d'instances uniformes, puis choisissez un type d'instance.

Pour créer un cluster Amazon EMR dans une Zone locale aide de l'AWS CLI

Utilisez la commande create-cluster, ainsi que l'ID de sous-réseau de la Zone locale, comme illustré dans l'exemple suivant. Remplacez subnet-22XXXX1234567 par l'ID de sous-réseau de la Zone locale et remplacez d'autres options si nécessaire. Pour de plus amples informations, veuillez consulter https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html.

```
aws emr create-cluster \
--name "Local Zones cluster" \
--release-label emr-5.29.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey,SubnetId=subnet-22XXXX1234567 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Configuration de Docker

Amazon EMR 6.x prend en charge Hadoop 3, qui permet au gestionnaire de nœuds YARN de lancer des conteneurs directement sur le Hôte du cluster EMR ou à l'intérieur d'un conteneur Docker. Les conteneurs Docker fournissent des environnements d'exécution personnalisés dans lesquels le code d'application s'exécute. L'environnement d'exécution personnalisé est isolé de l'environnement d'exécution du gestionnaire de nœuds YARN et d'autres applications.

Les conteneurs Docker peuvent inclure des bibliothèques spéciales utilisées par l'application et peuvent fournir différentes versions d'outils et de bibliothèques natifs, tels que R et Python. Vous pouvez utiliser des outils Docker familiers pour définir les bibliothèques et les dépendances d'exécution de vos applications.

Amazon EMR Les clusters 6.x sont configurés par défaut pour autoriser les applications YARN, telles que Spark, à s'exécuter à l'aide de conteneurs Docker. Pour personnaliser la configuration de votre conteneur, modifiez les options de support Docker définies dans les fichiers `yarn-site.xml` et `container-executor.cfg` disponibles dans le répertoire `/etc/hadoop/conf`. Pour plus d'informations sur chaque option de configuration et son utilisation, consultez Lancement d'applications à l'aide de conteneurs Docker.

Vous pouvez choisir d'utiliser Docker lorsque vous soumettez une tâche. Utilisez les variables suivantes pour spécifier l'environnement d'exécution Docker et l'image Docker.

- `YARN_CONTAINER_RUNTIME_TYPE=docker`
- `YARN_CONTAINER_RUNTIME_DOCKER_IMAGE={DOCKER_IMAGE_NAME}`

Lorsque vous utilisez des conteneurs Docker pour exécuter vos applications YARN, YARN télécharge l'image Docker que vous spécifiez lorsque vous soumettez votre tâche. Pour que YARN puisse résoudre cette image Docker, elle doit être configurée avec un registre Docker. Les options de configuration d'un registre Docker varient selon que vous déployez le cluster à l'aide d'un sous-réseau public ou privé.

Registres Docker

Un registre Docker est un système de stockage et de distribution pour les images Docker. Pour Amazon EMR 6.x, les registres Docker suivants peuvent être configurés :

- Docker Hub – Registre Docker public contenant plus de 100 000 images Docker populaires.
- Amazon ECR – Registre de conteneur Docker entièrement géré qui vous permet de créer vos propres images personnalisées et de les héberger dans une architecture hautement disponible et évolutive.

Considérations relatives au déploiement

Les registres Docker nécessitent un accès réseau à partir de chaque hôte du cluster. En effet, chaque hôte télécharge des images à partir du registre Docker lorsque votre application YARN est en cours d'exécution sur le cluster. Ces exigences de connectivité réseau peuvent limiter votre choix de registre Docker, selon que vous déployez votre cluster Amazon EMR dans un sous-réseau public ou privé.

Public subnet (Sous-réseau public)

Lorsque les clusters EMR sont déployés dans un sous-réseau public, les nœuds exécutant YARN NodeManager peuvent accéder directement à n'importe quel registre disponible sur Internet, y compris Docker Hub, comme illustré ci-dessous schéma.

Sous-réseau privé

Lorsque les clusters EMR sont déployés dans un sous-réseau privé, les nœuds exécutant le gestionnaire de nœuds YARN n'ont pas d'accès direct à Internet. Les images Docker peuvent être hébergées dans Amazon ECR et sont accessibles via AWS PrivateLink, comme illustré dans le diagramme suivant.

Pour plus d'informations sur l'utilisation d'AWS PrivateLink afin d'autoriser l'accès à Amazon ECR dans un scénario de sous-réseau privé, consultez Configuration d'AWS PrivateLink pour Amazon ECS, et Amazon ECR.

Configuration des registres Docker

Pour utiliser les registres Docker avec Amazon EMR, vous devez configurer Docker pour qu'il fasse confiance au registre spécifique que vous souhaitez utiliser afin de résoudre les images Docker. Les registres de confiance par défaut sont locaux (privés) et centos (sur Docker Hub public). Pour utiliser d'autres référentiels publics ou Amazon ECR, vous pouvez remplacer les paramètres `docker.trusted.registries` dans `/etc/hadoop/conf/container-executor.cfg` l'aide de l'API de classification EMR avec la clé de classification `container-executor`.

L'exemple suivant montre comment configurer le cluster pour approuver à la fois un référentiel public, nommé `your-public-repo`et un point de terminaison du registre ECR, `123456789123.dkr.ecr.us-east-1.amazonaws.com`. Si vous utilisez ECR, remplacez ce point de terminaison par votre point de terminaison ECR spécifique. Si vous utilisez Docker Hub, remplacez ce nom de référentiel par votre nom de référentiel.

```
[
  {
    "Classification": "container-executor",
    "Configurations": [
        {
            "Classification": "docker",
            "Properties": {
                "docker.trusted.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com",
                "docker.privileged-containers.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com"
            }
        }
    ]
  }
]
```

Pour lancer un Amazon EMR 6.0.0 Cluster avec cette configuration à l'aide de l' AWS Command Line Interface (AWS CLI), créez un fichier nommé `container-executor.json` avec le contenu du précédent de conteneur-exécuteur Configuration JSON. Ensuite, utilisez les commandes suivantes pour lancer le cluster.

```
export KEYPAIR=<Name of your Amazon EC2 key-pair>
export SUBNET_ID=<ID of the subnet to which to deploy the cluster>
export INSTANCE_TYPE=<Name of the instance type to use>
export REGION=<Region to which to deploy the cluster>

aws emr create-cluster \
    --name "EMR-6.0.0" \
    --region $REGION \
    --release-label emr-6.0.0 \
    --applications Name=Hadoop Name=Spark \
    --service-role EMR_DefaultRole \
    --ec2-attributes KeyName=$KEYPAIR,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=
$SUBNET_ID \
    --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=$INSTANCE_TYPE
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=$INSTANCE_TYPE \
    --configuration file://container-executor.json
```

Configuration de YARN pour accéder à Amazon ECR sur EMR 6.0.0 et versions antérieures

Si vous débutez avec Amazon ECR, suivez les instructions dans Mise en route avec Amazon ECR et vérifiez que vous avez accès à Amazon ECR de chaque instance de votre Amazon EMR du cluster.

Sur EMR 6.0.0 et versions antérieures, pour accéder Amazon ECR à l'aide de la commande Docker, vous devez d'abord générer des informations d'identification. Pour vérifier que YARN peut accéder aux images depuis Amazon ECR, utilisez la variable d'environnement de conteneur `YARN_CONTAINER_RUNTIME_DOCKER_CLIENT_CONFIG` pour transmettre une référence aux informations d'identification que vous avez générées.

Exécutez la commande suivante sur l'un des nœuds principaux pour générer la ligne de connexion correspondant à votre compte ECR.

```
aws ecr get-login --region us-east-1 --no-include-email
```

La commande `get-login` génère la commande CLI Docker correcte que vous devez exécuter pour créer les informations d'identification. Copiez et exécutez la sortie à partir de `get-login`.

```
sudo docker login -u AWS -p <password> https://<account-id>.dkr.ecr.us-east-1.amazonaws.com
```

Cette commande génère un fichier `config.json` dans le dossier `/root/.docker`. Copiez ce fichier dans HDFS afin que les tâches soumises au cluster puissent l'utiliser pour s'authentifier sur Amazon ECR.

Exécutez les commandes ci-dessous pour copier le fichier `config.json` dans votre répertoire personnel.

```
mkdir -p ~/.docker
sudo cp /root/.docker/config.json ~/.docker/config.json
sudo chmod 644 ~/.docker/config.json
```

Exécutez les commandes ci-dessous pour placer config.json dans HDFS afin qu'il puisse être utilisé par les tâches exécutées sur le cluster.

```
hadoop fs -put ~/.docker/config.json /user/hadoop/
```

YARN peut accéder à ECR en tant que registre d'images Docker et extraire les conteneurs lors de l'exécution de la tâche.

Après avoir configuré les registres Docker et YARN, vous pouvez exécuter des applications YARN à l'aide de conteneurs Docker. Pour plus d'informations, consultez Exécution des applications Spark avec Docker à l'aide d'Amazon EMR 6.0.0.

Dans EMR 6.1.0 et versions ultérieures, vous n'avez pas besoin de configurer manuellement l'authentification pour Amazon ECR. Si un Amazon ECR est détecté dans le `container-executor` la clé de classification, la Amazon ECR fonction d'authentification automatique active et YARN gère le processus d'authentification lorsque vous soumettez une tâche Spark avec une image ECR. Vous pouvez confirmer si l'authentification automatique est activé en cochant `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` dans le site de fil. L'authentification automatique est activée et le paramètre d'authentification YARN est défini sur `true` si le `docker.trusted.registries` contient une URL de registre ECR.

Conditions préalables pour l'utilisation de l'authentification automatique à Amazon ECR

- EMR version 6.1.0 ou ultérieure
- Le registre ECR inclus dans la configuration se trouve dans la même région que le cluster
- Rôle IAM avec autorisations pour obtenir un jeton d'autorisation et extraire une image

Reportez-vous à Configuration avec Amazon ECR pour plus d'informations.

Comment activer authentification automatique

Suivre Configuration des registres Docker (p. 81) pour définir un Amazon ECR en tant que registre approuvé, et assurez-vous que Amazon ECR et le cluster se trouvent dans la même région.

Pour activer cette fonctionnalité même lorsque le registre ECR n'est pas défini dans le registre approuvé, utilisez la classification de configuration pour définir `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` à `true`.

Comment désactiver authentification automatique

Par défaut, authentification automatique est désactivé si non Amazon ECR est détecté dans le registre approuvé.

Pour désactiver authentification automatique, même lorsque le Amazon ECR est défini dans le registre approuvé, utilisez la classification de configuration pour définir `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` à `false`.

Comment vérifier si l'authentification automatique est activé sur un du cluster

Sur le nœud principal, utilisez un éditeur de texte tel que `vi` pour afficher le contenu du fichier : `vi /etc/hadoop/conf.empty/yarn-site.xml`. Vérifiez la valeur de `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled`.

Utiliser le système de fichiers EMR (EMRFS)

Le système de fichiers EMR (EMRFS) est une mise en œuvre de HDFS que tous les clusters Amazon EMR utilisent pour la lecture et l'écriture des fichiers réguliers depuis Amazon EMR directement dans Amazon S3. EMRFS permet de stocker des données persistantes dans Amazon S3 en vue de les utiliser avec Hadoop, tout en fournissant des fonctionnalités telles qu'une lecture et un chiffrement cohérents des données.

La vue cohérente assure la vérification de la cohérence des listes et des lectures après écritures (pour les nouvelles demandes put) pour les objets dans Amazon S3. Le chiffrement de données vous permet de chiffrer les objets qu'EMRFS écrit dans Amazon S3 et permet à EMRFS de travailler avec les données chiffrées dans Amazon S3. Si vous utilisez les versions 4.8.0 ou ultérieures d'Amazon EMR, vous pouvez utiliser des configurations de sécurité pour configurer le chiffrement pour les objets EMRFS dans Amazon S3, ainsi que d'autres paramètres de chiffrement. Pour plus d'informations, consultez Options de chiffrement (p. 198), Si vous utilisez une version précédente version de Amazon EMR, vous pouvez configurer manuellement les paramètres de chiffrement. Pour plus d'informations, consultez Spécification du chiffrement Amazon S3 en utilisant les propriétés EMRFS (p. 102),

Lorsque vous utilisez les versions 5.10.0 ou ultérieures d'Amazon EMR, vous pouvez utiliser différents rôles IAM pour les demandes EMRFS à Amazon S3 en fonction des utilisateurs du cluster, des groupes ou de l'emplacement des données EMRFS dans Amazon S3. Pour plus d'informations, consultez Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234),

Rubriques

- Vue cohérente (p. 84)
- Autorisation d'accès aux données EMRFS dans Amazon S3 (p. 101)
- Spécification du chiffrement Amazon S3 en utilisant les propriétés EMRFS (p. 102)

Vue cohérente

La vue cohérente d'EMRFS est une option disponible lors de l'utilisation de Amazon EMR version 3.2.1 ou ultérieure. La vue cohérente permet aux clusters EMR de vérifier la cohérence et la cohérence read-after-write (lecture directe après écriture) de liste pour les objets Amazon S3 écrits par ou synchronisés avec le système EMRFS. La vue cohérente résout un problème qui peut découler du Modèle de cohérence des données Amazon S3. Par exemple, si vous ajoutez des objets à Amazon S3 en une seule opération et listez ensuite immédiatement les objets dans une opération ultérieure, la liste et l'ensemble d'objets traités peuvent être incomplets. C'est généralement un problème pour les clusters qui exécutent des étapes rapides et séquentielles en utilisant Amazon S3 en tant que magasin de données, telles que les pipelines de traitement de données ETL multi-étape.

Lorsque vous créez un cluster avec la vue cohérente activée, Amazon EMR utilise une base de données Amazon DynamoDB pour stocker les métadonnées d'objet et suivre la cohérence avec Amazon S3. Si la

vue cohérente détermine que Amazon S3 est incohérent au cours d'une opération de système de fichiers, il réessaie cette opération en fonction de règles que vous pouvez définir. Par défaut, la base de données DynamoDB a une capacité de lecture de 400 et une capacité d'écriture de 100. Vous pouvez configurer les paramètres de capacité de lecture/écriture en fonction du nombre d'objets dont EMRFS assure le suivi et le nombre de nœuds qui utilisent simultanément les métadonnées. Vous pouvez également configurer d'autres paramètres opérationnels et de base de données. L'utilisation de la vue cohérente entraîne des frais DynamoDB, qui sont généralement réduits, en plus des frais pour Amazon EMR. Pour de plus amples informations, veuillez consulter Tarification Amazon DynamoDB.

Avec la vue cohérente activée, EMRFS retourne le jeu des objets répertoriés dans les métadonnées EMRFS et de ceux retournés directement par Amazon S3 pour un chemin d'accès donné. Comme Amazon S3 est toujours la « source de vérité » pour les objets dans un chemin d'accès, EMRFS garantit que tout ce qui se trouve dans un chemin d'accès Amazon S3 spécifié est traité, que cela fasse l'objet d'un suivi ou non dans les métadonnées. Toutefois, la vue cohérente d'EMRFS garantit seulement la vérification de la cohérence des objets dans les dossiers dont vous effectuez le suivi.

Vous pouvez utiliser l'utilitaire EMRFS (`emrfs`) à partir de la ligne de commande du nœud principal pour effectuer des opérations sur les objets Amazon S3 qui sont suivis par la vue cohérente. Par exemple, vous pouvez importer, supprimer et synchroniser des objets Amazon S3 avec le magasin de métadonnées EMRFS. Pour plus d'informations sur l'utilitaire d'interface de ligne de commande d'EMRFS, consultez Référence de l'interface de ligne de commande EMRFS (p. 94).

Si vous supprimez des objets directement à partir de Amazon S3 qui sont suivis dans EMRFS des métadonnées, EMRFS traite l'objet comme incohérent et renvoie une exception après épuisement de nouvelles tentatives. Utilisez EMRFS pour supprimer des objets dans Amazon S3 qui sont suivis à l'aide de la vue cohérente. Autrement, vous pouvez utiliser la ligne de commande `emrfs` pour purger les entrées de métadonnées pour des objets qui ont été directement supprimés, ou vous pouvez synchroniser la vue cohérente avec Amazon S3 immédiatement après avoir supprimé les objets.

> **Note**
>
> Il est recommandé d'activer la durée de vie (TTL) pour le DynamoDB table créée par EMRFS. Le nom de la table par défaut est `EmrFSMetadata`. La durée de vie peut réduire les coûts en supprimant les volumes de données stockés qui ne sont plus nécessaires. Pour plus d'informations, consultez Activation de la durée de vie (TTL).

Rubriques

- Activation de la vue cohérente (p. 85)
- Comprendre comment la vue cohérente d'EMRFS effectue le suivi des objets dans Amazon S3 (p. 86)
- Logique des nouvelles tentatives (p. 87)
- Métadonnées de la vue cohérente EMRFS (p. 88)
- Configuration des notifications de cohérence pour CloudWatch et Amazon SQS (p. 90)
- Configuration de la vue cohérente (p. 91)
- Référence de l'interface de ligne de commande EMRFS (p. 94)

Activation de la vue cohérente

Vous pouvez activer Amazon S3 du chiffrement côté serveur ou vue cohérente pour EMRFS à l'aide de l' AWS Management Console, AWS CLI, ou le `emrfs-site` classification de configuration.

Pour configurer la vue cohérente à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).

3. Choisir les paramètres pour Étape 1 : Logiciels et étapes et Étape 2 : Matériel
4. Pour Étape 3 : Paramètres généraux du cluster, sous Options supplémentaires, choisissez Vue cohérente EMRFS.
5. Pour EMRFS Metadata store (Boutique de métadonnées EMRFS), saisissez le nom de votre magasin de métadonnées. La valeur par défaut est **`EmrFSMetadata`**. Si la table EmrFSMetadata n'existe pas, elle est créée pour vous dans DynamoDB.

   Note

   Amazon EMR ne supprime pas automatiquement les métadonnées EMRFS de DynamoDB lorsque le cluster est arrêté.
6. Pour Number of retries (Nombre de nouvelles tentatives), tapez une valeur entière. Si une incohérence est détectée, EMRFS essaie d'appeler Amazon S3 ce nombre de fois. La valeur par défaut est **`5`**.
7. Pour Retry period (in seconds) (Période de nouvelle tentative (en secondes)), tapez une valeur entière. Il s'agit du temps pendant lequel EMRFS attend entre les nouvelles tentatives. La valeur par défaut est **`10`**.

   Note

   Les nouvelles tentatives ultérieures utilisent une interruption exponentielle.

Pour lancer un cluster avec la vue cohérente activée à l'aide de l'AWS CLI

Nous vous recommandons d'installer la version actuelle de l'AWS CLI. Pour télécharger la dernière version, veuillez consulter https://aws.amazon.com//cli/.

- Note

  Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --instance-type m5.xlarge --instance-count 3 --emrfs
 Consistent=true \
--release-label emr-5.30.1 --ec2-attributes KeyName=myKey
```

Pour vérifier si la vue cohérente est activée à l'aide d'AWS Management Console

- Pour vérifier si la vue cohérente est activée dans la console, accédez à la Liste de clusters et sélectionnez le nom de votre cluster pour afficher Cluster Details (Détails de cluster). Le champ « EMRFS consistent view » a la valeur `Enabled` ou `Disabled`.

Pour vérifier si la vue cohérente est activée en examinant le fichier **`emrfs-site.xml`**

- Vous pouvez vérifier si la cohérence est activée en inspectant le fichier de configuration `emrfs-site.xml` sur le nœud maître du cluster. Si la valeur booléenne pour `fs.s3.consistent` est définie sur `true`, la vue cohérente est activée pour les opérations du système de fichiers impliquant Amazon S3.

Comprendre comment la vue cohérente d'EMRFS effectue le suivi des objets dans Amazon S3

EMRFS crée une vue cohérente des objets dans Amazon S3 en ajoutant des informations sur ces objets aux métadonnées EMRFS. EMRFS ajoute ces listes à ses métadonnées lorsque :

- un objet est écrit par EMRFS au cours d'une tâche Amazon EMR ;
- un objet est synchronisé avec les métadonnées EMRFS ou importé dans les métadonnées EMRFS à l'aide de l'interface de ligne de commande d'EMRFS.

Les objets lus par EMRFS ne sont pas automatiquement ajoutés aux métadonnées. Lorsqu'EMRFS supprime un objet, une liste demeure dans les métadonnées avec un état supprimé jusqu'à ce que cette liste soit purgée à l'aide de l'interface de ligne de commande d'EMRFS. Pour plus d'informations sur l'interface de ligne de commande, consultez Référence de l'interface de ligne de commande EMRFS (p. 94). Pour plus d'informations sur la purge des listes dans les métadonnées EMRFS, consultez Métadonnées de la vue cohérente EMRFS (p. 88).

Pour chaque opération Amazon S3, EMRFS vérifie les métadonnées pour obtenir des informations sur l'ensemble des objets dans la vue cohérente. Si EMRFS considère que Amazon S3 est incohérent au cours d'une de ces opérations, il réessaie l'opération selon les paramètres définis dans les propriétés de configuration `emrfs-site`. Une fois qu'EMRFS a épuisé les tentatives, il lève une exception `ConsistencyException` ou consigne l'exception et continue le flux de travail. Pour plus d'informations sur la logique des nouvelles tentatives, consultez Logique des nouvelles tentatives (p. 87). Vous pouvez trouver des exceptions `ConsistencyExceptions` dans vos journaux, par exemple :

- état de la liste : Non Amazon S3 objet pour l'élément de métadonnées `/S3_bucket/dir/object`
- getFileStatus (obtenir l'état du fichier) : Clé `dir/file` est présent dans les métadonnées, mais pas Amazon S3

Si vous supprimez un objet qui est suivi dans la vue cohérente d'EMRFS directement à partir de Amazon S3, EMRFS traite cet objet comme incohérent, car il reste répertorié dans les métadonnées tel qu'il figure dans Amazon S3. Si vos métadonnées cessent d'être synchronisées avec les objets qu'EMRFS suit dans Amazon S3, vous pouvez utiliser la sous-commande sync sur l'interface de ligne de commande d'EMRFS pour réinitialiser les métadonnées afin qu'elles tiennent compte de Amazon S3. Pour découvrir des incohérences entre les métadonnées et Amazon S3, utilisez le diff. Enfin, EMRFS n'a qu'une vue cohérente des objets référencés dans les métadonnées ; il peut y avoir d'autres objets dans le même Amazon S3 chemin qui n'est pas suivi. Quand EMRFS répertorie les objets dans un chemin d'accès Amazon S3, il renvoie le sur-ensemble des objets suivis dans les métadonnées et de ceux dans ce chemin d'accès Amazon S3.

Logique des nouvelles tentatives

EMRFS essaie de vérifier la cohérence des listes pour les objets suivis dans ses métadonnées pour un certain nombre de tentatives. La valeur par défaut est 5. Au cas où le nombre de nouvelles tentatives est dépassé, la tâche initiale retourne un échec, à moins que `fs.s3.consistent.throwExceptionOnInconsistency` ait la valeur `false`, auquel cas elle consignera uniquement les objets suivis comme incohérents. EMRFS utilise une stratégie de nouvelles tentatives d'interruption exponentielle par défaut, mais vous pouvez également la configurer comme une stratégie fixe. Les utilisateurs peuvent également réessayer pendant un certain temps avant de passer au reste de leur tâche sans lever d'exception. Ils peuvent y parvenir en définissant `fs.s3.consistent.throwExceptionOnInconsistency` sur `false`, `fs.s3.consistent.retryPolicyType` sur `fixed` et `fs.s3.consistent.retryPeriodSeconds` sur la valeur de leur choix. L'exemple suivant crée un cluster avec la cohérence activée, qui consigne les incohérences et définit un intervalle fixe de nouvelle tentative de 10 secondes :

Example Configuration de la période de nouvelle tentative sur une valeur fixe

```
aws emr create-cluster --release-label emr-5.30.1 \
--instance-type m5.xlarge --instance-count 1 \
--emrfs Consistent=true,Args=[fs.s3.consistent.throwExceptionOnInconsistency=false,
 fs.s3.consistent.retryPolicyType=fixed,fs.s3.consistent.retryPeriodSeconds=10] --ec2-
attributes KeyName=myKey
```

> **Note**
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

Pour plus d'informations, consultez Vue cohérente (p. 84),

Métadonnées de la vue cohérente EMRFS

La vue cohérente d'EMRFS assure la cohérence à l'aide d'une table DynamoDB pour suivre les objets dans Amazon S3 qui ont été synchronisés avec ou créés par le système EMRFS. Les métadonnées sont utilisées pour suivre toutes les opérations (lecture, écriture, mise à jour et copie) et aucun contenu réel n'y est stocké. Ces métadonnées sont utilisées pour vérifier si les objets ou les métadonnées provenant d'Amazon S3 correspondent à ce qui est prévu. Cette confirmation permet à EMRFS de contrôler la cohérence des listes et la cohérence des lectures après écritures pour les nouveaux objets qu'EMRFS écrit sur Amazon S3 ou pour les objets synchronisés avec EMRFS. Plusieurs clusters peuvent partager les mêmes métadonnées.

Comment ajouter des entrées aux métadonnées

Vous pouvez utiliser le `sync` ou `import` sous-commandes pour ajouter des entrées aux métadonnées. `sync` reflète simplement l'état des objets Amazon S3 dans un chemin d'accès lorsque la sous-commande `import` est utilisée strictement pour ajouter de nouvelles entrées aux métadonnées. Pour plus d'informations, consultez Référence de l'interface de ligne de commande EMRFS (p. 94),

Comment vérifier les différences entre les métadonnées et les objets dans Amazon S3

Pour vérifier les différences entre les métadonnées et Amazon S3, utilisez la sous-commande `diff` de l'interface de ligne de commande d'EMRFS. Pour plus d'informations, consultez Référence de l'interface de ligne de commande EMRFS (p. 94),

Comment savoir si les opérations de métadonnées sont limitées

EMRFS définit les limites de capacité de débit par défaut sur les métadonnées pour ses opérations de lecture et d'écriture à 500 et 100 unités, respectivement. Un grand nombre d'objets ou de compartiments peut entraîner le dépassement de cette capacité par les opérations, auquel cas DynamoDB va limiter les opérations. Par exemple, une application peut provoquer l'envoi par EMRFS d'un `ProvisionedThroughputExceededException` si vous effectuez une opération qui dépasse ces limites de capacité. Lors d'une limitation, l'outil d'interface de ligne de commande d'EMRFS tente de réécrire dans la table DynamoDB à l'aide d'une interruption exponentielle jusqu'à ce que l'opération s'achève ou qu'il atteigne la valeur maximale de nouvelles tentatives d'écriture d'objets d'Amazon EMR sur Amazon S3.

Vous pouvez configurer vos propres limites de capacité de débit. Cependant, DynamoDB a des limites de partition strictes de 3 000 unités de capacité en lecture (RCU) et de 1 000 unités de capacité en écriture (WCU) par seconde pour les opérations de lecture et d'écriture. Pour éviter `sync` des échecs causés par la limitation, nous vous recommandons de limiter le débit pour les opérations de lecture à moins de 3 000 RCU et les opérations d'écriture à moins de 1 000 WCU. Pour obtenir des instructions sur la définition des limites de capacité de débit personnalisées, consultez Configuration de la vue cohérente (p. 91).

Vous pouvez également consulter les métriques Amazon CloudWatch pour vos métadonnées EMRFS dans la console DynamoDB, qui vous indiquent le nombre de demandes de lecture et d'écriture limitées. Si le nombre de demandes limitées n'est pas nul, votre application peut éventuellement bénéficier d'une capacité grandissante de débit allouée pour les opérations de lecture ou d'écriture. Vous pouvez également réaliser une amélioration des performances si vous voyez que vos opérations approchent de la capacité de débit allouée maximale pour les lectures ou les écritures pendant une période de temps prolongée.

Caractéristiques de débit pour les opérations EMRFS notables

La valeur par défaut pour les opérations de lecture et d'écriture est de 400 et de 100 unités de capacité de débit, respectivement. Les caractéristiques de performance suivantes vous donnent une idée du

débit requis pour certaines opérations. Ces tests ont été effectués à l'aide d'un cluster `m3.large` à un seul nœud. Toutes les opérations étaient à thread unique. Les performances varient considérablement en fonction des caractéristiques d'application spécifiques et il peut être nécessaire d'expérimenter afin d'optimiser les opérations de système de fichiers.

Opération	Moyenne en lecture par seconde	Moyenne en écriture par seconde
create (objet)	26,79	6,70
delete (objet)	10,79	10,79
delete (répertoire contenant 1 000 objets)	21,79	338,40
getFileStatus (objet)	34,70	0 %
getFileStatus (répertoire)	19,96	0 %
listStatus (répertoire contenant 1 objet)	43,31	0 %
listStatus (répertoire contenant 10 objets)	44,34	0 %
listStatus (répertoire contenant 100 objets)	84,44	0 %
listStatus (répertoire contenant 1 000 objets)	308,81	0 %
listStatus (répertoire contenant 10 000 objets)	416,05	0 %
listStatus (répertoire contenant 100 000 objets)	823,56	0 %
listStatus (répertoire contenant 1 million d'objets)	882,36	0 %
mkdir (en continu pendant 120 secondes)	24,18	4,03
mkdir	12,59	0 %
rename (objet)	19,53	4,88
rename (répertoire contenant 1 000 objets)	23,22	339,34

Pour soumettre une étape qui purge les anciennes données à partir de votre magasin de métadonnées

Les utilisateurs peuvent supprimer des entrées particulières dans les métadonnées basées sur DynamoDB. Cela peut aider à réduire les coûts de stockage associés à la table. Les utilisateurs ont la possibilité de purger manuellement ou par programmation des entrées particulières à l'aide de la sous-commande `delete` de l'interface de ligne de commande d'EMRFS. Toutefois, si vous supprimez des entrées à partir des métadonnées, EMRFS n'effectue plus aucun contrôle de cohérence.

La purge par programmation à l'issue d'une tâche peut être effectuée en soumettant une étape finale à votre cluster qui exécute une commande sur l'interface de ligne de commande d'EMRFS. Par exemple,

tapez la commande suivante pour soumettre une étape à votre cluster afin de supprimer toutes les entrées de plus de deux jours.

```
aws emr add-steps --cluster-id j-2AL4XXXXXX5T9 --steps Name="emrfsCLI",Jar="command-
runner.jar",Args=["emrfs","delete","--time","2","--time-unit","days"]
{
    "StepIds": [
        "s-B12345678902"
    ]
}
```

Utilisez la valeur StepId retournée pour vérifier les journaux afin d'obtenir le résultat de l'opération.

Configuration des notifications de cohérence pour CloudWatch et Amazon SQS

Vous pouvez activer les métriques CloudWatch et les messages Amazon SQS dans EMRFS pour les problèmes de cohérence à terme d'Amazon S3.

CloudWatch

Lorsque les métriques CloudWatch sont activées, une métrique nommée Inconsistency (Incohérence) est poussée chaque fois qu'un appel d'API `FileSystem` échoue en raison de la cohérence à terme d'Amazon S3.

Pour afficher les métriques CloudWatch pour les problèmes de cohérence à terme d'Amazon S3

Pour afficher la métrique Inconsistency (Incohérence) dans la console CloudWatch, sélectionnez les métriques EMRFS, puis sélectionnez une paire JobFlowId/Metric (IDFluxTravail/Métrique) Name (Nom). Par exemple, `j-162XXXXXXM2CU ListStatus`, `j-162XXXXXXM2CU GetFileStatus`, et ainsi de suite sur.

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le Tableau de bord, dans la section Metrics (Métriques), choisissez EMRFS.
3. Dans le volet Job Flow Metrics (Métriques de flux de travail), sélectionnez une ou plusieurs paires JobFlowId (IDFluxTravail)/Metric Name (Nom de métrique). Une représentation graphique des métriques s'affiche dans la fenêtre ci-dessous.

Amazon SQS

Lorsque les notifications Amazon SQS sont activées, une file d'attente Amazon SQS avec le nom `EMRFS-Inconsistency-<jobFlowId>` est créée lors de l'initialisation d'EMRFS. Les messages Amazon SQS sont poussés dans la file d'attente lorsqu'un appel d'API `FileSystem` échoue en raison de la cohérence éventuelle de Amazon S3. Le message contient des informations telles que JobFlowId, l'API, une liste de chemins incohérents, une trace de la pile, etc. Les messages peuvent être lus à l'aide de la console Amazon SQS ou de la commande `read-sqs` d'EMRFS.

Pour gérer les messages Amazon SQS pour des problèmes de cohérence éventuelle d'Amazon S3.

Les messages Amazon SQS pour les problèmes de cohérence éventuelle d'Amazon S3 peuvent être lus à l'aide de l'interface de ligne de commande d'EMRFS. Pour lire les messages à partir d'une file d'attente Amazon SQS d'EMRFS, tapez la commande `read-sqs` et spécifiez un emplacement de sortie dans le système de fichiers local du nœud principal pour le fichier de sortie obtenu.

Vous pouvez également supprimer une file d'attente Amazon SQS d'EMRFS à l'aide de la commande `delete-sqs`.

1. Pour lire les messages à partir d'une file d'attente Amazon SQS, tapez la commande suivante. Remplacez *queuename* avec le nom du Amazon SQS file d'attente que vous avez configurée et remplacée */path/filename* avec le chemin d'accès au fichier de sortie :

```
emrfs read-sqs --queue-name queuename --output-file /path/filename
```

   Par exemple, pour lire et générer des messages Amazon SQS à partir de la file d'attente par défaut, tapez :

```
emrfs read-sqs --queue-name EMRFS-Inconsistency-j-162XXXXXXM2CU --output-file /path/
filename
```

   > **Note**
   >
   > Vous pouvez également utiliser les raccourcis `-q` et `-o` à la place de `--queue-name` et `--output-file` respectivement.

2. Pour supprimer une file d'attente Amazon SQS, tapez la commande suivante :

```
emrfs delete-sqs --queue-name queuename
```

   Par exemple, pour supprimer la file d'attente par défaut, tapez :

```
emrfs delete-sqs --queue-name EMRFS-Inconsistency-j-162XXXXXXM2CU
```

   > **Note**
   >
   > Vous pouvez également utiliser le raccourci `-q` à la place de `--queue-name`.

Configuration de la vue cohérente

Vous pouvez configurer d'autres paramètres pour la vue cohérente en les fournissant à l'aide des propriétés de configuration pour les propriétés `emrfs-site`. Par exemple, vous pouvez choisir un autre débit DynamoDB par défaut en fournissant les arguments suivants à l'interface de ligne de commande, à l'aide de l'option `--emrfs`, en utilisant la classification de configuration de site emrfs (versions 4.x et ultérieures d'Amazon EMR uniquement), ou une action d'amorçage pour configurer le fichier emrfs-site.xml sur le nœud principal :

Example Modification des valeurs de lecture et d'écriture de métadonnées par défaut au lancement du cluster

```
aws emr create-cluster --release-label emr-5.30.1 --instance-type m5.xlarge \
--emrfs Consistent=true,Args=[fs.s3.consistent.metadata.read.capacity=600,\
fs.s3.consistent.metadata.write.capacity=300] --ec2-attributes KeyName=myKey
```

Vous pouvez également utiliser le fichier de configuration suivant et l'enregistrer localement ou dans Amazon S3 :

```
[
    {
      "Classification": "emrfs-site",
      "Properties": {
        "fs.s3.consistent.metadata.read.capacity": "600",
        "fs.s3.consistent.metadata.write.capacity": "300"
      }
    }
```

```
]
```

Utilisez la configuration que vous avez créée avec la syntaxe suivante :

```
aws emr create-cluster --release-label emr-5.30.1 --applications Name=Hive \
--instance-type m5.xlarge --instance-count 2 --configurations file://./myConfig.json
```

> **Note**
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

Les options suivantes peuvent être définies à l'aide des configurations ou des arguments `--emrfs` de l'AWS CLI. Pour plus d'informations sur ces arguments, consultez la AWS CLI Command Reference.

Propriétés **`emrfs-site.xml`** pour la vue cohérente

Propriété	Valeur par défaut : ""	Description :
`fs.s3.consistent`	**`false`**	Lorsque la valeur est **`true`**, cette propriété configure EMRFS pour utiliser DynamoDB afin d'assurer la cohérence.
`fs.s3.consistent.retryPolicyType`	**`exponential`**	Cette propriété identifie la stratégie à utiliser lors d'une nouvelle tentative pour des problèmes de cohérence. Les options comprennent : exponential, fixed et none.
`fs.s3.consistent.retryPeriodSeconds`	**`1`**	Cette propriété définit la durée d'attente entre les tentatives de relance de cohérence.
`fs.s3.consistent.retryCount`	**`10`**	Cette propriété définit le nombre maximal de nouvelles tentatives lorsqu'une incohérence est détectée.
`fs.s3.consistent.throwExceptionOnInconsistency`	**`true`**	Cette propriété détermine s'il convient de lever ou de consigner une exception de cohérence. Lorsque la valeur est **`true`**, une exception `ConsistencyException` est levée.
`fs.s3.consistent.metadata.autoCreate`	**`true`**	Lorsque la valeur est **`true`**, cette propriété permet la création automatique de tables de métadonnées.
`fs.s3.consistent.metadata.etag.verification.enabled`	**`true`**	Avec Amazon EMR 5.29.0, cette propriété est activée par défaut. Lorsqu'elle est activée, EMRFS utilise S3 ETags pour vérifier que

Propriété	Valeur par défaut : ""	Description :
		les objets en cours de lecture correspondent à la dernière version disponible. Cette fonctionnalité est utile pour les cas d'utilisation de lecture après mise à jour dans lesquels des fichiers sur S3 sont remplacés tout en conservant le même nom. Cette fonctionnalité de vérification ETag ne fonctionne actuellement pas avec S3 Select.
`fs.s3.consistent.metadata.tableName`	**`EmrFSMetadata`**	Cette propriété spécifie le nom de la table de métadonnées dans DynamoDB.
`fs.s3.consistent.metadata.read.capacity`	**`500`**	Cette propriété spécifie la capacité de lecture DynamoDB à mettre en service lorsque la table de métadonnées est créée.
`fs.s3.consistent.metadata.write.capacity`	**`100`**	Cette propriété spécifie la capacité d'écriture DynamoDB à mettre en service lorsque la table de métadonnées est créée.
`fs.s3.consistent.fastList`	**`true`**	Lorsque la valeur est **`true`**, cette propriété utilise plusieurs fils pour répertorier un répertoire (si nécessaire). La cohérence doit être activée pour pouvoir utiliser cette propriété.
`fs.s3.consistent.fastList.prefetchMetadata`	**`false`**	Lorsque la valeur est **`true`**, cette propriété permet l'extraction préalable de métadonnées pour les répertoires contenant plus de 20 000 éléments.
`fs.s3.consistent.notification.CloudWatch`	**`false`**	Lorsque la valeur est **`true`**, les métriques CloudWatch sont activées pour les appels d'API de système de fichiers qui échouent en raison de problèmes de cohérence à terme d'Amazon S3.
`fs.s3.consistent.notification.SQS`	**`false`**	Lorsque la valeur est **`true`**, les notifications de cohérence éventuelle font l'objet d'un push vers une file d'attente Amazon SQS.

Propriété	Valeur par défaut : ""	Description :
`fs.s3.consistent.notification.SQS.queueName`	**`EMRFS-Inconsistency-<jobFlowId>`**	La modification de cette propriété vous permet de spécifier votre propre nom de file d'attente SQS pour les messages concernant les problèmes de cohérence à terme d'Amazon S3.
`fs.s3.consistent.notification.SQS.customMsg`	**`none`**	Cette propriété vous permet de spécifier des informations personnalisées incluses dans les messages SQS concernant les problèmes de cohérence à terme d'Amazon S3. Si une valeur n'est pas spécifiée pour cette propriété, le champ correspondant dans le message est vide.
`fs.s3.consistent.dynamodb.endpoint`	**`none`**	Cette propriété vous permet de spécifier un point de terminaison DynamoDB personnalisé pour vos métadonnées de vue cohérente.

Référence de l'interface de ligne de commande EMRFS

L'interface de ligne de commande EMRFS est installée par défaut sur tous les nœuds maîtres du cluster créé à l'aide de Amazon EMR 3.2.1 ou version supérieure. Vous pouvez utiliser l'interface de ligne de commande EMRFS pour gérer les métadonnées pour la vue cohérente.

> **Note**
>
> La commande emrfs est uniquement prise en charge avec l'émulation de terminal VT100. Cependant, elle peut fonctionner avec les autres modes d'émulateur de terminal.

Commande emrfs de niveau supérieur

La commande emrfs de niveau supérieur prend en charge la structure suivante.

```
emrfs [describe-metadata | set-metadata-capacity | delete-metadata | create-metadata | \
list-metadata-stores | diff | delete | sync | import ] [options] [arguments]
```

Spécifiez des [options], avec ou sans arguments, comme décrit dans le tableau suivant. Pour les [options] spécifiques à des sous-commandes (`describe-metadata`, `set-metadata-capacity`, etc.), consultez chaque sous-commande ci-dessous.

[options] pour emrfs

Option	Description :	: obligatoire.
`-a` *`AWS_ACCESS_KEY_ID`* `| --access-key` *`AWS_ACCESS_KEY_ID`*	La clé d'accès AWS que vous utilisez pour écrire des objets sur Amazon S3 et pour créer ou accéder à un magasin de métadonnées dans DynamoDB. Par défaut, *`AWS_ACCESS_KEY_ID`* est défini sur la clé d'accès utilisée pour créer le cluster.	Non.

Option	Description :	: obligatoire.
`-s` *`AWS_SECRET_ACCESS_KEY`* `| --secret-key` *`AWS_SECRET_ACCESS_KEY`*	La clé secrète AWS associée à la clé d'accès que vous utilisez pour écrire des objets dans Amazon S3 et pour créer un magasin de métadonnées ou y accéder dans DynamoDB. Par défaut, *`AWS_SECRET_ACCESS_KEY`* est défini sur la clé secrète associée à la clé d'accès utilisée pour créer le cluster.	Non.
`-v | --verbose`	Détaillez le résultat.	Non.
`-h | --help`	Affiche le message d'aide pour la commande `emrfs` avec une instruction d'utilisation.	Non.

Sous commande describe-metadata emrfs

[options] pour describe-metadata emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* `| --metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.

Example Exemple describe-metadata emrfs

L'exemple suivant décrit la table de métadonnées par défaut.

```
$ emrfs describe-metadata
EmrFSMetadata
  read-capacity: 400
  write-capacity: 100
  status: ACTIVE
  approximate-item-count (6 hour delay): 12
```

Sous-commande set-metadata-capacity emrfs

[options] pour set-metadata-capacity emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* `| --metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.
`-r` *`READ_CAPACITY`* `| --read-capacity` *`READ_CAPACITY`*	La capacité de débit de lecture demandée pour la table de métadonnées. Si l'icône *`READ_CAPACITY`* n'est pas fourni, la valeur par défaut est `400`.	Non.
`-w` *`WRITE_CAPACITY`* `| --write-capacity` *`WRITE_CAPACITY`*	La capacité de débit d'écriture demandée pour la table de métadonnées. Si l'icône *`WRITE_CAPACITY`* n'est pas fourni, la valeur par défaut est `100`.	Non.

Example Exemple set-metadata-capacity emrfs

L'exemple suivant définit la capacité de débit de lecture sur `600` et la capacité d'écriture sur `150` pour une table de métadonnées nommée `EmrMetadataAlt`.

```
$ emrfs set-metadata-capacity --metadata-name EmrMetadataAlt  --read-capacity 600 --write-
capacity 150
  read-capacity: 400
  write-capacity: 100
  status: UPDATING
  approximate-item-count (6 hour delay): 0
```

Sous commande delete-metadata emrfs

[options] pour delete-metadata emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* | `--metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.

Example Exemple delete-metadata emrfs

L'exemple suivant supprime la table de métadonnées par défaut.

```
$ emrfs delete-metadata
```

Sous commande create-metadata emrfs

[options] pour create-metadata emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* | `--metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.
`-r` *`READ_CAPACITY`* | `--read-capacity` *`READ_CAPACITY`*	La capacité de débit de lecture demandée pour la table de métadonnées. Si l'icône *`READ_CAPACITY`* n'est pas fourni, la valeur par défaut est `400`.	Non.
`-w` *`WRITE_CAPACITY`* | `--write-capacity` *`WRITE_CAPACITY`*	La capacité de débit d'écriture demandée pour la table de métadonnées. Si l'icône *`WRITE_CAPACITY`* n'est pas fourni, la valeur par défaut est `100`.	Non.

Example Exemple create-metadata emrfs

L'exemple suivant crée une table de métadonnée nommée `EmrFSMetadataAlt`.

```
$ emrfs create-metadata -m EmrFSMetadataAlt
Creating metadata: EmrFSMetadataAlt
EmrFSMetadataAlt
  read-capacity: 400
  write-capacity: 100
```

```
  status: ACTIVE
  approximate-item-count (6 hour delay): 0
```

Sous-commande list-metadata-stores emrfs

La sous-commande emrfs list-metadata-stores n'a aucune [option].

Example Exemple list-metadata-stores

L'exemple suivant répertorie vos tables de métadonnées.

```
$ emrfs list-metadata-stores
  EmrFSMetadata
```

Sous-commande diff emrfs

[options] pour diff emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* | `--metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.
`s3://s3Path`	Le chemin d'accès au compartiment Amazon S3 à comparer à la table des métadonnées. Synchronisation de compartiments de façon récursive.	: oui

Example Exemple diff emrfs

L'exemple suivant compare la table de métadonnées par défaut dans un compartiment Amazon S3.

```
$ emrfs diff s3://elasticmapreduce/samples/cloudfront
BOTH | MANIFEST ONLY | S3 ONLY
DIR elasticmapreduce/samples/cloudfront
DIR elasticmapreduce/samples/cloudfront/code/
DIR elasticmapreduce/samples/cloudfront/input/
DIR elasticmapreduce/samples/cloudfront/logprocessor.jar
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-14.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-15.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-16.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-17.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-18.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-19.WxYz1234
DIR elasticmapreduce/samples/cloudfront/input/XABCD12345678.2009-05-05-20.WxYz1234
DIR elasticmapreduce/samples/cloudfront/code/cloudfront-loganalyzer.tgz
```

Sous commande delete emrfs

[options] pour delete emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* | `--metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.

Option	Description :	: obligatoire.
s3://s3Path	Le chemin d'accès au compartiment Amazon S3 que vous suivez pour une vue cohérente. Synchronisation de compartiments de façon récursive.	: oui
`-t` *TIME* | `--time` *TIME*	L'heure d'expiration (interprétée à l'aide de l'argument d'unité de temps). Toutes les entrées de métadonnées antérieures à la *TIME* sont supprimés pour le compartiment spécifié.	
`-u` *UNIT* | `--time-unit` *UNIT*	La mesure utilisée pour interpréter l'argument temps (nanosecondes, microsecondes, millisecondes, secondes, minutes, heures ou jours). Si aucun argument n'est spécifié, la valeur par défaut est `days`.	
`--read-consumption` *READ_CONSUMPTION*	Le montant requis de débit de lecture disponible utilisé pour l'opération delete. Si l'icône *READ_CONSUMPTION* n'est pas spécifié, la valeur par défaut est `400`.	Non.
`--write-consumption` *WRITE_CONSUMPTION*	Le montant requis de débit d'écriture disponible utilisé pour l'opération delete. Si l'icône *WRITE_CONSUMPTION* n'est pas spécifié, la valeur par défaut est `100`.	Non.

Example Exemple delete emrfs

L'exemple suivant supprime tous les objets dans un compartiment Amazon S3 depuis les métadonnées de suivi pour une vue cohérente.

```
$ emrfs delete s3://elasticmapreduce/samples/cloudfront
entries deleted: 11
```

Sous commande import emrfs

[options] pour import emrfs

Option	Description :	: obligatoire.
`-m` *METADATA_NAME* | `--metadata-name` *METADATA_NAME*	*METADATA_NAME* est le nom du DynamoDB table de métadonnées. Si l'icône *METADATA_NAME* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.
s3://s3Path	Le chemin d'accès au compartiment Amazon S3 que vous suivez pour une vue cohérente. Synchronisation de compartiments de façon récursive.	: oui
`--read-consumption` *READ_CONSUMPTION*	Le montant requis de débit de lecture disponible utilisé pour l'opération delete. Si l'icône *READ_CONSUMPTION* n'est pas spécifié, la valeur par défaut est `400`.	Non.
`--write-consumption` *WRITE_CONSUMPTION*	Le montant requis de débit d'écriture disponible utilisé pour l'opération delete. Si l'icône *WRITE_CONSUMPTION* n'est pas spécifié, la valeur par défaut est `100`.	Non.

Example Exemple import emrfs

L'exemple suivant importe tous les objets dans un compartiment Amazon S3 avec les métadonnées de suivi pour une vue cohérente. Toutes les clés inconnus sont ignorées.

```
$ emrfs import s3://elasticmapreduce/samples/cloudfront
```

Sous-commande sync emrfs

[options] pour sync emrfs

Option	Description :	: obligatoire.
`-m` *`METADATA_NAME`* | `--metadata-name` *`METADATA_NAME`*	*`METADATA_NAME`* est le nom du DynamoDB table de métadonnées. Si l'icône *`METADATA_NAME`* n'est pas fourni, la valeur par défaut est `EmrFSMetadata`.	Non.
`s3://s3Path`	Le chemin d'accès au compartiment Amazon S3 que vous suivez pour une vue cohérente. Synchronisation de compartiments de façon récursive.	: oui
`--read-consumption` *`READ_CONSUMPTION`*	Le montant requis de débit de lecture disponible utilisé pour l'opération delete. Si l'icône *`READ_CONSUMPTION`* n'est pas spécifié, la valeur par défaut est `400`.	Non.
`--write-consumption` *`WRITE_CONSUMPTION`*	Le montant requis de débit d'écriture disponible utilisé pour l'opération delete. Si l'icône *`WRITE_CONSUMPTION`* n'est pas spécifié, la valeur par défaut est `100`.	Non.

Example Exemple commande sync emrfs

L'exemple suivant importe tous les objets dans un compartiment Amazon S3 avec les métadonnées de suivi pour une vue cohérente. Toutes les clés inconnues sont supprimées.

```
$ emrfs sync s3://elasticmapreduce/samples/cloudfront
Synching samples/cloudfront                                    0 added | 0 updated | 0
 removed | 0 unchanged
Synching samples/cloudfront/code/                              1 added | 0 updated | 0
 removed | 0 unchanged
Synching samples/cloudfront/                                   2 added | 0 updated | 0
 removed | 0 unchanged
Synching samples/cloudfront/input/                             9 added | 0 updated | 0
 removed | 0 unchanged
Done synching s3://elasticmapreduce/samples/cloudfront         9 added | 0 updated | 1
 removed | 0 unchanged
creating 3 folder key(s)
folders written: 3
```

Sous-commande read-sqs emrfs

[options] pour read-sqs emrfs

Option	Description :	: obligatoire.
`-q` *`QUEUE_NAME`* | `--queue-name` *`QUEUE_NAME`*	*`QUEUE_NAME`* est le nom du Amazon SQS file d'attente configurée dans `emrfs-site.xml`. La valeur par défaut est **`EMRFS-Inconsistency-<jobFlowId>`**.	: oui
`-o` *`OUTPUT_FILE`* | `--output-file` *`OUTPUT_FILE`*	*`OUTPUT_FILE`* est le chemin d'accès au fichier de sortie sur le système de fichiers local du nœud principal. Les messages lus depuis la file d'attente sont écrits dans ce fichier.	: oui

Sous commande delete-sqs emrfs

[options] pour delete-sqs emrfs

Option	Description :	: obligatoire.
`-q` *`QUEUE_NAME`* | `--queue-name` *`QUEUE_NAME`*	*`QUEUE_NAME`* est le nom du Amazon SQS file d'attente configurée dans `emrfs-site.xml`. La valeur par défaut est **`EMRFS-Inconsistency-<jobFlowId>`**.	: oui

Soumission de commandes CLI EMRFS comme étapes

L'exemple suivant montre comment utiliser l'utilitaire `emrfs` sur le nœud principal en tirant parti de l'AWS CLI ou de l'API ainsi que `command-runner.jar` pour exécuter la commande `emrfs` en tant qu'étape. L'exemple utilise le kit SDK AWS pour Python (Boto3) pour ajouter une étape à un cluster qui ajoute des objets dans un Amazon S3 vers la table de métadonnées EMRFS par défaut.

```
import boto3
from botocore.exceptions import ClientError


def add_emrfs_step(command, bucket_url, cluster_id, emr_client):
    """
    Add an EMRFS command as a job flow step to an existing cluster.

    :param command: The EMRFS command to run.
    :param bucket_url: The URL of a bucket that contains tracking metadata.
    :param cluster_id: The ID of the cluster to update.
    :param emr_client: The Boto3 Amazon EMR client object.
    :return: The ID of the added job flow step. Status can be tracked by calling
             the emr_client.describe_step() function.
    """
    job_flow_step = {
        'Name': 'Example EMRFS Command Step',
        'ActionOnFailure': 'CONTINUE',
        'HadoopJarStep': {
            'Jar': 'command-runner.jar',
            'Args': [
                '/usr/bin/emrfs',
                command,
                bucket_url
            ]
```

```
        }
    }

    try:
        response = emr_client.add_job_flow_steps(
            JobFlowId=cluster_id, Steps=[job_flow_step])
        step_id = response['StepIds'][0]
        print(f"Added step {step_id} to cluster {cluster_id}.")
    except ClientError:
        print(f"Couldn't add a step to cluster {cluster_id}.")
        raise
    else:
        return step_id


def usage_demo():
    emr_client = boto3.client('emr')
    # Assumes the first waiting cluster has EMRFS enabled and has created metadata
    # with the default name of 'EmrFSMetadata'.
    cluster = emr_client.list_clusters(ClusterStates=['WAITING'])['Clusters'][0]
    add_emrfs_step(
        'sync', 's3://elasticmapreduce/samples/cloudfront', cluster['Id'], emr_client)


if __name__ == '__main__':
    usage_demo()
```

Vous pouvez utiliser le `step_id` valeur renvoyée pour vérifier les journaux pour le résultat de l'opération.

Autorisation d'accès aux données EMRFS dans Amazon S3

Par défaut, le rôle EMR pour EC2 détermine les autorisations d'accès aux données EMRFS dans Amazon S3. Les stratégies IAM qui sont attachées à ce rôle s'appliquent, quel que soit l'utilisateur ou le groupe qui effectuent la demande via le système EMRFS. La valeur par défaut est `EMR_EC2_DefaultRole`. Pour plus d'informations, consultez Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).

À partir de la version 5.10.0 d'Amazon EMR, vous pouvez utiliser une configuration de sécurité pour spécifier les rôles IAM pour EMRFS. Cela vous permet de personnaliser les autorisations pour les demandes EMRFS à Amazon S3 pour les clusters ayant plusieurs utilisateurs. Vous pouvez spécifier différents rôles IAM pour différents utilisateurs et groupes, et pour différents emplacements de compartiment Amazon S3 en fonction du préfixe dans Amazon S3. Quand EMRFS envoie une demande à Amazon S3 qui correspond à des utilisateurs, des groupes ou des emplacements que vous spécifiez, le cluster utilise le rôle correspondant que vous spécifiez au lieu du rôle EMR pour EC2. Pour plus d'informations, consultez Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234),

Sinon, si les exigences de votre solution Amazon EMR ne sont pas couvertes par les rôles IAM pour EMRFS, vous pouvez définir une classe de fournisseur d'informations d'identification personnalisées, ce qui vous permet de personnaliser l'accès aux données EMRFS dans Amazon S3.

Création d'un fournisseur d'informations d'identification personnalisées pour les données EMRFS dans Amazon S3

Pour créer un fournisseur d'informations d'identification personnalisées, vous implémentez les classes AWSCredentialsProvider et Configurable Hadoop.

Pour obtenir une explication détaillée de l'approche, consultez Analyse sécurisée des données depuis un autre compte AWS avec EMRFS dans le blog Big Data AWS. Le billet de blog inclut un didacticiel qui parcourt le processus de bout en bout, depuis la création des rôles IAM au lancement du cluster. Il

fournit aussi un exemple de code Java qui implémente la classe fournisseur d'informations d'identification personnalisées.

La procédure de base est la suivante :

Pour spécifier un fournisseur d'informations d'identification personnalisées

1. Créez une classe fournisseur d'informations d'identification personnalisées compilée comme fichier JAR.
2. Exécutez un script comme action d'amorçage pour copier le fichier JAR du fournisseur d'informations d'identification personnalisées sur l'emplacement `/usr/share/aws/emr/emrfs/auxlib` du nœud principal du cluster. Pour plus d'informations sur les actions d'amorçage, consultez (Facultatif) Création d'actions d'amorçage pour installer des logiciels supplémentaires.
3. Personnalisez la classification `emrfs-site` pour spécifier la classe que vous implémentez dans le fichier JAR. Pour plus d'informations sur la spécification des objets de configuration pour personnaliser les applications, consultez Configuration des applications dans le Amazon EMR Guide de version.

   L'exemple suivant illustre une commande `create-cluster` qui lance un cluster Hive avec les paramètres de configuration courants et qui inclut également :

   - Une action d'amorçage qui exécute le script, *`copy_jar_file.sh`*, enregistré sur *`mybucket`* dans Amazon S3.
   - Une classification `emrfs-site` qui spécifie un fournisseur d'informations d'identification personnalisées défini dans le fichier JAR comme *`MyCustomCredentialsProvider`*

   > **Note**
   >
   > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --applications Name=Hive \
--bootstrap-actions '[{"Path":"s3://mybucket/copy_jar_file.sh","Name":"Custom
 action"}]' \
--ec2-attributes '{"KeyName":"MyKeyPair","InstanceProfile":"EMR_EC2_DefaultRole",\
"SubnetId":"subnet-xxxxxxxx","EmrManagedSlaveSecurityGroup":"sg-xxxxxxxx",\
"EmrManagedMasterSecurityGroup":"sg-xxxxxxxx"}' \
--service-role EMR_DefaultRole --enable-debugging --release-label emr-5.30.1 \
--log-uri 's3n://my-emr-log-bucket/' --name 'test-awscredentialsprovider-emrfs' \
--instance-type=m5.xlarge --instance-count 3  \
--configurations '[{"Classification":"emrfs-site",\
"Properties":{"fs.s3.customAWSCredentialsProvider":"MyAWSCredentialsProviderWithUri"},\
"Configurations":[]}]'
```

Spécification du chiffrement Amazon S3 en utilisant les propriétés EMRFS

> **Important**
>
> A compter de la version Amazon EMR 4.8.0, vous pouvez utiliser les configurations de sécurité pour appliquer les paramètres de chiffrement plus facilement et avec plus d'options. Nous vous recommandons d'utiliser les configurations de sécurité. Pour plus d'informations, consultez la section Configuration du chiffrement des données (p. 180). Les instructions de console décrites dans cette section sont disponibles pour les versions antérieures à la version 4.8.0. Si vous utilisez l'AWS CLI pour configurer le chiffrement Amazon S3 à la fois dans la configuration de cluster

et dans une configuration de sécurité dans des versions ultérieures, la configuration de sécurité prévaut.

Lorsque vous créez un cluster, vous pouvez spécifier le chiffrement côté serveur (SSE) ou le chiffrement côté client (CSE) pour les données EMRFS dans Amazon S3 à l'aide de la console ou de `emrfs-site` des propriétés de classification via le AWS CLI ou SDK EMR. Amazon S3 SSE et CSE sont mutuellement exclusifs ; vous pouvez choisir entre les deux, mais pas les deux.

Pour les instructions AWS CLI, reportez-vous à la section appropriée pour votre type de chiffrement ci-dessous.

Pour spécifier les options de chiffrement EMRFS à l'aide de la AWS Management Console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Choisissez une Release (Version) 4.7.2 ou antérieure.
4. Choisissez d'autres options pour Software and Steps (Logiciel et étapes) selon les besoins de votre application, puis cliquez sur Next (Suivant).
5. Choisissez les paramètres dans les volets Hardware (Matériel) et General Cluster Settings (Paramètres de cluster généraux) selon les besoins de votre application.
6. Dans le volet Security (Sécurité), sous Authentication and encryption (Authentification et chiffrement), sélectionnez l'option S3 Encryption (with EMRFS) (Chiffrement S3 (avec EMRFS)) à utiliser.

   Note

   S3 server-side encryption with KMS Key Management (Chiffrement côté serveur S3 avec gestion de clé KMS) (SSE-KMS) n'est pas disponible lorsque vous utilisez les versions 4.4 ou antérieures d'Amazon EMR.

   - Si vous choisissez une option qui utilise la AWS Key Management (Gestion de clés AWS), choisissez un AWS KMS Key ID (ID de clé KMS AWS). Pour plus d'informations, consultez Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement EMRFS (p. 103),
   - Si vous choisissez le S3 client-side encryption with custom materials provider (Chiffrement côté client S3 avec fournisseur de matériaux personnalisé), spécifiez le Class name (Nom de classe) et l'JAR location (emplacement du JAR). Pour plus d'informations, consultez Chiffrement Amazon S3 côté client (p. 105),
7. Choisissez d'autres options selon les besoins de votre application, puis choisissez Create Cluster (Créer un cluster).

Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement EMRFS

La clé de chiffrement AWS KMS doit être créée dans la même région que votre instance de cluster Amazon EMR et les compartiments Amazon S3 utilisés avec EMRFS. Si la clé que vous spécifiez figure dans un compte différent de celui que vous utilisez pour configurer un cluster, vous devez spécifier la clé au moyen de son nom ARN.

Le rôle pour le profil d'instance Amazon EC2 doit avoir les autorisations nécessaires pour utiliser la clé CMK que vous spécifiez. Le rôle par défaut pour le profil d'instance dans Amazon EMR est `EMR_EC2_DefaultRole`. Si vous utilisez un autre rôle pour le profil d'instance ou si vous utilisez les rôles IAM pour les demandes EMRFS adressées à Amazon S3, vérifiez que chaque rôle est ajouté en tant qu'utilisateur clé, selon les besoins. Cela donne au rôle les autorisations nécessaires pour utiliser la CMK. Pour de plus amples informations, veuillez consulter Utilisation de stratégies de clé dans le AWS Key Management Service Developer Guide et Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).

Vous pouvez utiliser AWS Management Console pour ajouter votre profil d'instance ou le profil d'instance EC2 à la liste des utilisateurs de clés pour la clé CMK AWS KMS spécifiée, ou vous pouvez utiliser AWS CLI ou un kit SDK AWS pour attacher une stratégie de clé appropriée.

La procédure ci-dessous explique comment ajouter le profil d'instance EMR par défaut, `EMR_EC2_DefaultRole`, en tant qu'utilisateur de clés via AWS Management Console. Elle suppose que vous avez déjà créé une clé CMK. Pour créer une nouvelle clé CMK, consultez Création de clés dans le AWS Key Management Service Developer Guide.

Pour ajouter le profil d'instance EC2 pour Amazon EMR à la liste des utilisateurs de clés de chiffrement

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Sélectionnez l'alias de la clé CMK à modifier.
4. Sur la page de détails de la clé, sous Key Users (Utilisateurs de clés), choisissez Add (Ajouter).
5. Dans la boîte de dialogue Ajouter des utilisateurs clés sélectionnez le rôle approprié. Le nom du rôle par défaut est `EMR_EC2_DefaultRole`.
6. Choisissez Ajouter.

Chiffrement Amazon S3 côté serveur

Lorsque vous configurez le chiffrement Amazon S3 côté serveur, Amazon S3 chiffre les données au niveau de l'objet lors de l'écriture des données sur le disque et les déchiffre lorsque quelqu'un y accède. Pour plus d'informations sur le chiffrement SSE, consultez Protection des données à l'aide d'un chiffrement côté serveur dans le Amazon Simple Storage Service Manuel du développeur.

Vous pouvez choisir entre deux systèmes de gestion des clés différents quand vous spécifiez SSE dans Amazon EMR :

- SSE-S3 – Amazon S3 gère les clés pour vous.
- SSE-KMS – Vous utilisez une clé principale du client (CMK) AWS KMS configurée avec des stratégies adaptées à Amazon EMR. Pour plus d'informations sur les exigences relatives aux clés pour Amazon EMR, consultez Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement (p. 203).

Le chiffrement SSE avec des clés fournies par le client (SSE-C) n'est pas disponible avec Amazon EMR.

Pour créer un cluster avec le chiffrement SSE-S3 activé à l'aide de l'AWS CLI

- Saisissez la commande suivante.

```
aws emr create-cluster --release-label emr-4.7.2 or earlier \
--instance-count 3 --instance-type m5.xlarge --emrfs Encryption=ServerSide
```

Vous pouvez également activer SSE-S3 en définissant la propriété fs.s3.enableServerSideEncryption sur vrai dans les propriétés `emrfs-site`. Voir l'exemple pour SSE-KMS ci-dessous et omettez la propriété pour l'ID de clé.

Pour créer un cluster avec le chiffrement SSE-KMS activé à l'aide de l'AWS CLI

> **Note**
>
> SSE-KMS est uniquement disponible dans Amazon EMR version 4.5.0 et versions ultérieures.

- Saisissez les informations suivantes AWS CLI pour créer un cluster avec SSE-KMS, où *keyID* est un AWS KMS clé principale client (CMK), par exemple, *a4567b8-9900-12ab-1234-123a45678901*:

```
aws emr create-cluster --release-label emr-4.7.2 or earlier --instance-count 3 \
--instance-type m5.xlarge --use-default-roles \
--emrfs Encryption=ServerSide,Args=[fs.s3.serverSideEncryption.kms.keyId=keyId]
```

--OR--

Saisissez la commande suivante de l'AWS CLI à l'aide de la classification `emrfs-site` et fournissez un fichier de configuration JSON avec le contenu indiqué similaire à `myConfig.json` dans l'exemple ci-dessous :

```
aws emr create-cluster --release-label emr-4.7.2 or earlier --instance-count 3
 --instance-type m5.xlarge --applications Name=Hadoop --configurations file://
myConfig.json --use-default-roles
```

Exemple de contenu de myConfig.json :

```
[
  {
    "Classification":"emrfs-site",
    "Properties": {
       "fs.s3.enableServerSideEncryption": "true",
       "fs.s3.serverSideEncryption.kms.keyId":"a4567b8-9900-12ab-1234-123a45678901"
    }
  }
]
```

Propriétés de configuration pour SSE-S3 et SSE-KMS

Ces propriétés peuvent être configurées à l'aide de la classification de configuration `emrfs-site`. SSE-KMS est uniquement disponible dans Amazon EMR version 4.5.0 et versions ultérieures.

Propriété	Valeur par défaut : ""	Description :
`fs.s3.enableServerSideEncryption`	**`false`**	Lorsque la valeur est **`true`**, les objets stockés dans Amazon S3 sont chiffrés à l'aide du chiffrement côté serveur. Si aucune clé n'est spécifiée, SSE-S3 est utilisé.
`fs.s3.serverSideEncryption.kms.keyId`	**`n/a`**	Spécifie un ID ou un ARN de clé AWS KMS. Si une clé est spécifiée, SSE-KMS est utilisé.

Chiffrement Amazon S3 côté client

Avec le chiffrement Amazon S3 côté client, le chiffrement et le déchiffrement Amazon S3 se produisent dans le client EMRFS, sur votre cluster. Les objets sont chiffrés avant d'être chargés dans Amazon S3 et déchiffrés après leur téléchargement. Le fournisseur que vous spécifiez fournit la clé de chiffrement que le client utilise. Le client peut utiliser les clés fournies par AWS KMS (CSE-KMS) ou une classe Java personnalisée qui fournit la clé principale côté client (CSE-C). Les détails de chiffrement sont légèrement

différents entre CSE-KMS et CSE-C, selon le fournisseur spécifié et les métadonnées de l'objet déchiffré ou chiffré. Pour de plus amples informations sur ces différences, veuillez consulter Protection des données via le chiffrement côté client dans le Amazon Simple Storage Service Manuel du développeur.

> **Note**
>
> Le chiffrement côté client Amazon S3 garantit uniquement le chiffrement des données EMRFS échangées avec Amazon S3. Les données des volumes d'instance du cluster ne sont pas toutes chiffrées. En outre, comme Hue n'utilise pas EMRFS, les objets écrits sur Amazon S3 à l'aide de l'explorateur de fichiers S3 Hue ne sont pas chiffrés.

Pour spécifier CSE-KMS pour les données EMRFS dans Amazon S3 à l'aide de l'AWS CLI

- Tapez la commande suivante et remplacez *MyKMSKeyID* avec l'ID de clé ou l'ARN du AWS KMS Clé CMK à utiliser :

```
aws emr create-cluster --release-label emr-4.7.2 or earlier
--emrfs Encryption=ClientSide,ProviderType=KMS,KMSKeyId=MyKMSKeyId
```

Création d'un fournisseur de clés personnalisé

Lorsque vous créez un fournisseur de clés personnalisé, l'application doit mettre en œuvre l'interface EncryptionMaterialsProvider, qui est disponible dans le kit AWS SDK for Java version 1.11.0 ou ultérieure. Cette mise en œuvre peut utiliser n'importe quelle stratégie pour fournir les supports de chiffrement. Vous pouvez, par exemple, choisir de fournir des supports de chiffrement statique ou d'intégrer un système de gestion de clés plus complexe.

L'algorithme de chiffrement utilisé pour les supports de chiffrement personnalisés doit être AES/GCM/NoPadding.

La classe EncryptionMaterialsProvider récupère les supports de chiffrement par contexte de chiffrement. Amazon EMR renseigne les informations de contexte de chiffrement au moment de l'exécution pour aider le mandataire à déterminer les supports de chiffrement corrects à renvoyer.

Example Exemple : Utilisation d'un fournisseur de clés personnalisé pour le chiffrement Amazon S3 avec EMRFS

Lorsqu'Amazon EMR extrait les supports de chiffrement de la classe EncryptionMaterialsProvider pour effectuer le chiffrement, EMRFS renseigne éventuellement l'argument materialsDescription avec deux champs : l'URI Amazon S3 de l'objet et l'ID JobFlowId du cluster, qui peuvent être utilisés par la classe EncryptionMaterialsProvider pour renvoyer les supports de chiffrement de façon sélective.

Par exemple, le fournisseur peut renvoyer des clés différentes pour différents préfixes d'URI Amazon S3. La description des supports de chiffrement renvoyés est finalement stockée avec l'objet Amazon S3 plutôt que la valeur materialsDescription qui est générée par EMRFS et transmise au fournisseur. Lors du déchiffrement d'un objet Amazon S3, la description des supports de chiffrement est transmise à la classe EncryptionMaterialsProvider, afin qu'elle puisse, à nouveau, renvoyer de façon sélective la clé correspondante pour déchiffrer l'objet.

Une mise en œuvre de référence de la classe EncryptionMaterialsProvider est fournie ci-dessous. Un autre fournisseur personnalisé, EMRFSRSAEncryptionMaterialsProvider, est disponible à partir de GitHub.

```
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.EncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.KMSEncryptionMaterials;
import org.apache.hadoop.conf.Configurable;
import org.apache.hadoop.conf.Configuration;
```

```
import java.util.Map;

/**
 * Provides KMSEncryptionMaterials according to Configuration
 */
public class MyEncryptionMaterialsProviders implements EncryptionMaterialsProvider,
 Configurable{
  private Configuration conf;
  private String kmsKeyId;
  private EncryptionMaterials encryptionMaterials;

  private void init() {
    this.kmsKeyId = conf.get("my.kms.key.id");
    this.encryptionMaterials = new KMSEncryptionMaterials(kmsKeyId);
  }

  @Override
  public void setConf(Configuration conf) {
    this.conf = conf;
    init();
  }

  @Override
  public Configuration getConf() {
    return this.conf;
  }

  @Override
  public void refresh() {

  }

  @Override
  public EncryptionMaterials getEncryptionMaterials(Map<String, String>
 materialsDescription) {
    return this.encryptionMaterials;
  }

  @Override
  public EncryptionMaterials getEncryptionMaterials() {
    return this.encryptionMaterials;
  }
}
```

Spécification d'un fournisseur de matériaux personnalisés à l'aide de la AWS CLI

Pour utiliser le AWS CLI, réussissez le `Encryption`, `ProviderType`, `CustomProviderClass`, et `CustomProviderLocation` aux arguments `emrfs` option.

```
aws emr create-cluster --instance-type m5.xlarge --release-label emr-4.7.2 or earlier
 --emrfs Encryption=ClientSide,ProviderType=Custom,CustomProviderLocation=s3://mybucket/
myfolder/provider.jar,CustomProviderClass=classname
```

L'affectation à `ClientSide` de la valeur `Encryption` rend possible le chiffrement côté client. `CustomProviderClass` est le nom de votre objet EncryptionMaterialsProvider et `CustomProviderLocation` est l'emplacement local ou Amazon S3 à partir duquel Amazon EMR copie `CustomProviderClass` sur chaque nœud du cluster et le place dans le chemin de classe.

Spécification d'un fournisseur de matériaux personnalisés à l'aide d'un SDK

Pour utiliser un kit SDK, vous pouvez définir la propriété `fs.s3.cse.encryptionMaterialsProvider.uri` pour télécharger la classe

EncryptionMaterialsProvider personnalisée que vous stockez dans Amazon S3 sur chaque nœud de votre cluster. Vous configurez cela dans le fichier `emrfs-site.xml` avec CSE activé et l'emplacement correct du fournisseur personnalisé.

Par exemple, dans le AWS SDK for Java utilisant RunJobFlowRequest, votre code peut ressembler à ce qui suit :

```
<snip>
  Map<String,String> emrfsProperties = new HashMap<String,String>();
      emrfsProperties.put("fs.s3.cse.encryptionMaterialsProvider.uri","s3://mybucket/
MyCustomEncryptionMaterialsProvider.jar");
      emrfsProperties.put("fs.s3.cse.enabled","true");
      emrfsProperties.put("fs.s3.consistent","true");

 emrfsProperties.put("fs.s3.cse.encryptionMaterialsProvider","full.class.name.of.EncryptionMaterialsPro

  Configuration myEmrfsConfig = new Configuration()
      .withClassification("emrfs-site")
      .withProperties(emrfsProperties);

  RunJobFlowRequest request = new RunJobFlowRequest()
   .withName("Custom EncryptionMaterialsProvider")
   .withReleaseLabel("emr-5.30.1")
   .withApplications(myApp)
   .withConfigurations(myEmrfsConfig)
   .withServiceRole("EMR_DefaultRole")
   .withJobFlowRole("EMR_EC2_DefaultRole")
   .withLogUri("s3://myLogUri/")
   .withInstances(new JobFlowInstancesConfig()
    .withEc2KeyName("myEc2Key")
    .withInstanceCount(2)
    .withKeepJobFlowAliveWhenNoSteps(true)
    .withMasterInstanceType("m5.xlarge")
    .withSlaveInstanceType("m5.xlarge")
   );

  RunJobFlowResult result = emr.runJobFlow(request);
</snip>
```

EncryptionMaterialsProvider personnalisé avec arguments

Vous devrez peut-être transmettre des arguments directement au fournisseur. Pour ce faire, vous pouvez utiliser la classification de configuration `emrfs-site` avec les propriétés définies comme arguments personnalisés. Un exemple de configuration est indiqué ci-dessous, qui est enregistré dans un fichier, `myConfig.json` :

```
[
    {
      "Classification": "emrfs-site",
      "Properties": {
        "myProvider.arg1":"value1",
     "myProvider.arg2":"value2"
      }
    }
 ]
```

En utilisant la commande `create-cluster` à partir de l'AWS CLI, vous pouvez utiliser l'option `--configurations` pour spécifier le fichier comme indiqué ci-dessous :

```
aws emr create-cluster --release-label emr-5.30.1 --instance-type m5.xlarge
 --instance-count 2 --configurations file://myConfig.json --emrfs
```

```
Encryption=ClientSide,CustomProviderLocation=s3://mybucket/myfolder/
myprovider.jar,CustomProviderClass=classname
```

Propriétés `emrfs-site.xml` pour le chiffrement Amazon S3 côté client

Propriété	Valeur par défaut : ""	Description :
`fs.s3.cse.enabled`	**`false`**	Lorsque la valeur est **`true`**, les objets EMRFS stockés dans Amazon S3 sont chiffrés à l'aide du chiffrement côté client.
`fs.s3.cse.encryptionMaterialsProvider.uri`	**N/A**	S'applique lors de l'utilisation de supports de chiffrement personnalisés. L'URI Amazon S3 où se trouve le fichier JAR avec EncryptionMaterialsProvider. Lorsque vous fournissez cet URI, Amazon EMR télécharge automatiquement le fichier JAR sur tous les nœuds du cluster.
`fs.s3.cse.encryptionMaterialsProvider`	**N/A**	Le chemin de classe `EncryptionMaterialsProvider` utilisé avec le chiffrement côté client. Lorsque vous utilisez CSE-KMS, spécifiez `com.amazon.ws.emr.hadoop.fs.cse.KMSEnc`
`fs.s3.cse.materialsDescription.enabled`	**`false`**	Lorsqu'elle est définie sur `true`, elle remplit l'attribut materialsDescription des objets chiffrés avec l'URI Amazon S3 de l'objet et l'ID JobFlowId. Définissez sur `true` lors de l'utilisation de supports de chiffrement personnalisés.
`fs.s3.cse.kms.keyId`	**N/A**	S'applique lors de l'utilisation de CSE-KMS. La valeur de l'ID de la clé, ARN, ou l'alias de la CMK AWS KMS utilisée pour le chiffrement.
`fs.s3.cse.cryptoStorageMode`	**`ObjectMetadata`**	Mode de stockage Amazon S3. Par défaut, la description des informations de chiffrement est stockée dans les métadonnées de l'objet. Vous pouvez également stocker la description dans un fichier d'instructions. Les valeurs valides sont ObjectMetadata et InstructionFile. Pour plus d'informations, consultez Chiffrement de données côté client avec le kit AWS SDK pour Java et Amazon S3.

Control Cluster Termination

Lorsque vous créez un cluster à l'aide d'Amazon EMR, vous pouvez choisir de créer un cluster transitoire qui s'arrête automatiquement une fois les étapes terminées, ou vous pouvez créer un cluster de longue durée qui continue à s'exécuter jusqu'à ce que vous l'arrêtiez volontairement. Lorsqu'un cluster est arrêté, toutes les instances Amazon EC2 dans le cluster s'arrêtent et les données dans le stockage d'instance et dans les volumes EBS ne sont plus disponibles et ne sont pas récupérables. Comprendre et gérer la résiliation du cluster est critique pour l'élaboration d'une stratégie pour gérer et conserver les données par l'écriture sur Amazon S3 et la répartition des coûts. Pour plus d'informations sur la résiliation manuelle d'un cluster, consultez Arrêter un cluster (p. 382).

Lorsque vous utilisez la résiliation automatique, le cluster démarre, exécute toute action d'amorçage que vous spécifiez, puis exécute généralement des données d'entrée, des étapes pour traiter les données, puis produit et enregistre la sortie. Lorsque les étapes se terminent, Amazon EMR résilie automatiquement les instances Amazon EC2 de cluster. Il s'agit d'un modèle efficace pour un cluster qui exécute une tâche de traitement périodique, par exemple une exécution de traitement de données quotidiennes. L'arrêt automatique du cluster aide à assurer que seul le temps nécessaire pour traiter les données vous est facturé. Pour plus d'informations sur ces étapes, consultez Work with Steps Using the AWS CLI and Console (p. 418).

Le cluster démarre de la même manière lorsqu'il est de longue durée. Vous pouvez spécifier des étapes comme vous le feriez avec un cluster qui se résilie automatiquement, mais le cluster continue à s'exécuter et d'accumuler des frais une fois les étapes terminées. Ce modèle est efficace lorsque vous avez besoin d'interroger des données de manière interactive ou automatique, ou d'interagir continuellement avec des applications de big data hébergées sur le cluster. Le traitement régulier est également efficace si vous avez un ensemble de données si grand ou si fréquent qu'il est inefficace de lancer de nouveaux clusters et de charger des données à chaque fois. Vous pouvez activer la protection contre la résiliation sur les clusters de longue durée pour éviter l'arrêt accidentel. Vous pouvez également tirer parti des fonctionnalités telles que le dimensionnement automatique et les parcs d'instances pour dimensionner le cluster de manière dynamique afin d'équilibrer les performances et le coût en réponse à la charge de travail. Pour de plus amples informations, veuillez consulter Dimensionnement des ressources de cluster (p. 384) et Configuration de parcs d'instances (p. 149).

Cette section décrit comment fonctionnent la protection contre la résiliation et l'arrêt automatique, et la façon dont ils interagissent les uns avec les autres, avec d'autres Amazon EMR fonctionnalités et avec d'autres données.

Rubriques

- Configuring a Cluster to Auto-Terminate or Continue (p. 110)
- Using Termination Protection (p. 111)

Configuring a Cluster to Auto-Terminate or Continue

Par défaut, les clusters que vous créez à l'aide de la console ou de l'AWS CLI continuent de s'exécuter jusqu'à ce que vous les arrêtez. Pour qu'un cluster se résilie après l'exécution des étapes, vous devez activer l'arrêt automatique. En revanche, les clusters que vous lancez à l'aide de l'API EMR disposent de l'arrêt automatique activé par défaut.

Pour désactiver l'arrêt automatique à l'aide de l'API EMR

- Lorsque vous utilisez l'action RunJobFlow pour créer un cluster, définissez la propriété KeepJobFlowAliveWhenNoSteps sur `true`.

Pour activer l'arrêt automatique à l'aide des Options rapides dans la AWS Management Console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Stop execution (Arrêter l'exécution).
4. Choisissez d'autres paramètres selon les besoins de votre application, puis choisissez Create Cluster (Créer un cluster).

Pour activer l'arrêt automatique à l'aide des Options avancées dans la AWS Management Console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Sous Ajouter des étapes (facultatif), sélectionnez Arrêter automatiquement le cluster après la fin de la dernière étape.
5. Choisissez d'autres paramètres selon les besoins de votre application, puis choisissez Create Cluster (Créer un cluster).

Pour activer l'arrêt automatique à l'aide de l'AWS CLI

- Spécifiez le paramètre `--auto-terminate` quand vous utilisez la commande `create-cluster` pour créer un cluster transitoire.

  L'exemple suivant illustre l'utilisation du paramètre `--auto-terminate`. Vous pouvez saisir la commande suivante et remplacer *`myKey`* avec le nom de votre paire de clés EC2.

  > Note
  >
  > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.30.1 \
--applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey \
--steps Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,\
Args=[-f,s3://mybucket/scripts/pigscript.pig,-p,\
INPUT=s3://mybucket/inputdata/,-p,OUTPUT=s3://mybucket/outputdata/,\
$INPUT=s3://mybucket/inputdata/,$OUTPUT=s3://mybucket/outputdata/]
--instance-type m5.xlarge --instance-count 3 --auto-terminate
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez la Référence AWS CLI.

Using Termination Protection

Lorsque la protection contre la résiliation est activée sur un cluster de longue durée, vous pouvez toujours résilier le cluster, mais vous devez explicitement supprimer la protection contre la résiliation du cluster pour pouvoir l'arrêter. Cela permet d'assurer que les instances EC2 ne sont pas résiliées par accident ou par erreur. Cette protection contre la résiliation est particulièrement utile si votre cluster contient des données stockées sur des disques locaux que vous avez besoin de récupérer avant que ces instances soient résiliées. Vous pouvez activer la protection de la résiliation lorsque vous créez un cluster et vous pouvez modifier le paramètre sur un cluster en cours d'exécution.

Lorsque la protection de la résiliation est activée, l'action `TerminateJobFlows` dans l'API Amazon EMR ne fonctionne pas. Les utilisateurs ne peuvent pas résilier le cluster à l'aide de cette API ou de la commande `terminate-clusters` dans l'AWS CLI. L'API retourne une erreur et l'interface de ligne de commande se ferme avec un code de retour non nul. Lorsque vous utilisez la console Amazon EMR pour arrêter un cluster, vous êtes invité à désactiver la protection de la résiliation par une étape supplémentaire.

> **Warning**
>
> La protection contre l'arrêt ne garantit pas que les données seront conservées en cas d'erreur humaine ou en cas de solution—par exemple, si une commande de redémarrage est exécutée à partir de la ligne de commande alors que vous êtes connecté à l'instance avec SSH, si une application ou un script s'exécutant sur l'instance publie une commande de redémarrage ou si l'API Amazon EC2 ou Amazon EMR est utilisée pour désactiver la protection contre l'arrêt. Même avec la protection contre l'arrêt activée, les données enregistrées sur le stockage d'instance, y compris les données HDFS, peuvent être perdues. Écrivez les sorties de données à des emplacements Amazon S3 et créez des stratégies de sauvegarde, conformément à vos exigences en matière de continuité des opérations.

La protection de la résiliation n'affecte pas votre capacité à dimensionner des ressources de cluster en utilisant l'une des actions suivantes :

- Resizing a cluster manually using the AWS Management Console or AWS CLI. For more information, see Redimensionnement manuel d'un cluster en cours d'exécution (p. 410).
- Removing instances from a core or task instance group using a scale-in policy with automatic scaling. For more information, see Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401).
- Removing instances from an instance fleet by reducing target capacity. For more information, see Options de parc d'instances (p. 150).

Termination Protection and Amazon EC2

Un cluster Amazon EMR avec la protection de la résiliation activée comporte l'attribut `disableAPITermination` défini pour toutes les instances Amazon EC2 du cluster. Si une demande de résiliation provient de l'Amazon EMR, et des paramètres Amazon EMR et Amazon EC2 pour un conflit d'instance, le paramètre Amazon EMR remplace le paramètre Amazon EC2. Par exemple, si vous utilisez Amazon EC2 console à enable protection de terminaison sur un Amazon EC2 dans un cluster qui a une protection de terminaison disabled, lorsque vous utilisez le Amazon EMR console, AWS CLI commandes pour Amazon EMR, ou Amazon EMR API pour mettre fin au cluster, Amazon EMR ensembles `DisableApiTermination` vers `false` et termine l'instance avec d'autres instances.

> **Important**
>
> Si une instance est créée dans le cadre d'un cluster Amazon EMR avec la protection de la résiliation, et l'API Amazon EC2 ou les commandes AWS CLI sont utilisées pour modifier l'instance afin que `DisableApiTermination` soit `false`, puis l'API Amazon EC2 ou les commandes AWS CLI exécutent l'action `TerminateInstances`, l'instance Amazon EC2 se résilie.

Termination Protection and Unhealthy YARN Nodes

Amazon EMR vérifie périodiquement l'état Apache Hadoop YARN des nœuds principaux s'exécutant sur des instances Amazon EC2 de noyau et de tâche dans un cluster. L'état de santé est signalé par le Service de vérification de l'état NodeManager. Si un nœud indique `UNHEALTHY`, le contrôleur d'instance Amazon EMR met le nœud sur la liste noire et ne permet pas d'allouer des conteneurs YARN à celui-ci jusqu'à ce qu'il redevienne sain. La raison la plus commune pour des nœuds non sains s'explique par le fait que l'utilisation des disques dépasse 90 %. Pour plus d'informations sur l'identification des nœuds non sains et leur récupération, consultez Erreurs de ressource (p. 443).

Si le nœud reste `UNHEALTHY` pendant plus de 45 minutes, Amazon EMR prend l'action suivante en fonction de l'état de la protection de la résiliation.

Protection contre la mise hors service	Résultat:
Activé (Recommandé)	L'instance Amazon EC2 demeure dans un état de liste noire et continue de contribuer à la capacité de cluster. Vous pouvez vous connecter à l'instance Amazon EC2 pour la configuration et la récupération des données, et redimensionner votre cluster pour ajouter de la capacité. Pour plus d'informations, consultez Erreurs de ressource (p. 443),
Désactivé	L'instance Amazon EC2 est résiliée. Amazon EMR provisionne une nouvelle instance en fonction du nombre d'instances spécifié dans le groupe d'instances ou de la capacité cible pour les parcs d'instances. Si tous les nœuds principaux sont `UNHEALTHY` pendant plus de 45 minutes, le cluster se résilie avec un rapport d'état `NO_SLAVES_LEFT`. Important Les données HDFS peuvent être perdues si une instance de noyau est résiliée à cause d'un état non sain. Si le nœud a stocké des blocs qui n'ont pas été répliqués à d'autres nœuds, ces blocs sont perdus, ce qui peut entraîner une perte de données. Nous vous recommandons d'utiliser la protection de la résiliation afin de pouvoir vous connecter à des instances et récupérer les données si nécessaire.

Termination Protection, Auto-Termination, and Step Execution

Le paramètres d'arrêt automatique est prioritaire sur la protection de la résiliation. Si les deux sont activés, lorsque les étapes finissent de s'exécuter, le cluster se résilie au lieu de procéder à un état d'attente.

Lorsque vous soumettez des étapes à un cluster, vous pouvez définir la propriété `ActionOnFailure` pour déterminer ce qui se produit si l'étape ne peuvent pas terminer son exécution en raison d'une erreur. Les valeurs possibles pour ce paramètre sont `TERMINATE_CLUSTER` (`TERMINATE_JOB_FLOW` avec des versions antérieures), `CANCEL_AND_WAIT`, et `CONTINUE`. Pour plus d'informations, voir Work with Steps Using the AWS CLI and Console (p. 418).

Si une étape qui est configurée avec la `ActionOnFailure` valeur `CANCEL_AND_WAIT` échoue, si l'arrêt automatique est activé, le cluster se résilie sans exécuter d'étapes suivantes.

Si une étape qui est configurée avec la `ActionOnFailure` valeur `TERMINATE_CLUSTER` échoue, utilisez la table de paramètres ci-dessous pour déterminer le résultat.

ActionOnFailure	Arrêt automatique	Protection contre la mise hors service	Résultat:
`TERMINATE_CLUSTER`	Enabled	Désactivé	Le cluster se résilie

ActionOnFailure	Arrêt automatique	Protection contre la mise hors service	Résultat:
	Enabled	Enabled	Le cluster se résilie
	Désactivé	Enabled	Le cluster continue
	Désactivé	Désactivé	Le cluster se résilie

Termination Protection and Spot Instances

La protection de la résiliation Amazon EMR n'empêche pas la résiliation d'une instance Spot Amazon EC2 lorsque le prix Spot dépasse le prix Spot maximum.

Configuring Termination Protection When You Launch a Cluster

Vous pouvez activer ou désactiver la protection contre l'arrêt lorsque vous lancez un cluster à l'aide de la console, de l'AWS CLI ou de l'API.

Par défaut, le paramètre de protection de la résiliation dépend de la façon dont vous lancez le cluster :

- Amazon EMR Console Quick Options—Termination Protection is disabled by default.
- Amazon EMR Console Advanced Options—Termination Protection is enabled by default.
- AWS CLI `aws emr create-cluster`—Termination Protection is disabled unless `--termination-protected` is specified.
- Amazon EMR API RunJobFlow command—Termination Protection is disabled unless the `TerminationProtected` boolean value is set to `true`.

Pour activer ou désactiver la protection de la résiliation lors de la création d'un cluster à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Pour Étape 3 : Paramètres généraux du cluster, sous Options générales assurez-vous Protection de terminaison est sélectionné pour l'activer, ou effacer la sélection pour la désactiver.

5. Choisissez d'autres paramètres selon les besoins de votre application, choisissez Next (Suivant), puis terminer la configuration de votre cluster.

Pour activer ou désactiver la protection de la résiliation lors de la création d'un cluster à l'aide de l'AWS CLI

- À l'aide de l'AWS CLI, vous pouvez lancer un cluster avec la protection de la résiliation activée en utilisant la commande `create-cluster` avec le paramètre `--termination-protected`. La protection de la résiliation est désactivée par défaut.

  L'exemple suivant crée un cluster avec la protection de la résiliation activée :

  > Note
  >
  > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "TerminationProtectedCluster" --release-label emr-5.30.1
 \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --termination-protected
```

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Configuring Termination Protection for Running Clusters

Vous pouvez configurer la protection contre l'arrêt pour un cluster en cours d'exécution à l'aide de la console ou de l'AWS CLI.

Pour activer ou désactiver la protection de la résiliation d'un cluster en cours d'exécution à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Sur la page Clusters, choisissez le Nom de votre cluster.
3. Dans l'onglet Summary (Récapitulatif), pour la Termination protection (Protection de la résiliation), choisissez Change (Modifier).
4. Pour activer la protection de la résiliation, choisissez On (Activée). Pour désactiver la protection de la résiliation, choisissez Off (Désactivée). Ensuite, sélectionnez la coche verte pour confirmer.

Pour activer ou désactiver la protection de la résiliation d'un cluster en cours d'exécution à l'aide de l'AWS CLI

- Pour activer la protection de la résiliation sur un cluster en cours d'exécution à l'aide de l'AWS CLI, utilisez la commande `modify-cluster-attributes` avec le paramètre `--termination-protected`. Pour la désactiver, utilisez le paramètre `--no-termination-protected`.

  L'exemple suivant permet la protection de terminaison sur le cluster avec ID *`j-3KVTXXXXXX7UG`*:

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --termination-protected
  ```

  L'exemple suivant désactive la protection de la résiliation sur le même cluster :

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-
  protected
  ```

Utilisation des AMI Amazon Linux dans Amazon EMR

Amazon EMR utilise une Amazon Machine Image (AMI) Amazon Linux pour initialiser les instances Amazon EC2 lorsque vous créez et lancez un cluster. L'AMI intègre le système d'exploitation Amazon Linux, d'autres logiciels et les configurations requises pour chaque instance pour héberger vos applications de cluster.

Par défaut, lorsque vous utilisez un cluster, Amazon EMR utilise une AMI Amazon Linux par défaut qui est spécifiquement créée pour la version Amazon EMR que vous utilisez. Lorsque vous utilisez les versions 5.7.0 ou ultérieures d'Amazon EMR, vous pouvez choisir de spécifier une AMI Amazon Linux personnalisée au lieu de l'AMI Amazon Linux par défaut pour Amazon EMR. Une AMI personnalisée vous permet de chiffrer le volume du périphérique racine et de personnaliser les applications et les configurations comme alternative à l'utilisation d'actions d'amorçage.

Amazon EMR attache automatiquement un volume SSD d'usage général Amazon EBS en tant que dispositif racine pour toutes les AMI. L'utilisant d'une AMI à support EBS améliore la performance. Les coûts EBS sont calculés au prorata du nombre d'heures en fonction des frais Amazon EBS mensuels pour les volumes gp2 de la région où s'exécute le cluster. Par exemple, le coût EBS par heure pour le volume racine sur chaque instance de cluster dans une région qui facture 0,10 $/Go/mois est d'environ 0,00139 $ par heure (0,10 $/Go/mois divisé par 30 jours et divisé par 24 h fois 10 Go). Que vous utilisiez l'AMI Amazon Linux par défaut ou une AMI Amazon Linux personnalisée, vous pouvez spécifier la taille du volume du périphérique racine EBS de 10 à 100 Gio.

Pour en savoir plus sur les AMI Amazon Linux, consultez Amazon Machine Images (AMI). Pour plus d'informations sur le stockage d'instances pour les instances Amazon EMR, consultez Stockage d'instance (p. 136).

Rubriques

- À l'aide de l'AMI Amazon Linux par défaut pour Amazon EMR (p. 117)
- Utilisation d'une image AMI personnalisée (p. 118)
- Spécification de la taille du volume du périphérique racine Amazon EBS (p. 123)

À l'aide de l'AMI Amazon Linux par défaut pour Amazon EMR

Chaque version Amazon EMR utilise une AMI Amazon Linux différente pour Amazon EMR, sauf si vous spécifiez une AMI personnalisée. L'AMI par défaut est basée sur l'AMI Amazon Linux la plus à jour et disponible au moment de la publication d'Amazon EMR. L'AMI est testée pour la compatibilité avec des applications de big data et les fonctionnalités Amazon EMR incluses avec cette version.

Chaque version Amazon EMR est « verrouillée » à la version AMI Amazon Linux pour conserver la compatibilité. Cela signifie que la même version d'AMI Amazon Linux est utilisée pour une version d'Amazon EMR, même lorsque de nouvelles AMI Amazon Linux deviennent disponibles. Pour cette raison, nous vous conseillons d'utiliser la version d'Amazon EMR la plus récente (actuellement 5.30.1), sauf si vous avez besoin d'une version antérieure pour la compatibilité et que vous n'êtes pas en mesure de migrer.

Si vous devez utiliser une version antérieure d'Amazon EMR pour des raisons de compatibilité, nous vous conseillons d'utiliser la version la plus récente d'une série. Par exemple, si vous devez utiliser la série 5.12, utilisez 5.12.2 plutôt que 5.12.0 ou 5.12.1. Si une nouvelle version devient disponible dans une série, envisagez de migrer vos applications vers cette nouvelle version.

Gestion de mises à jour logicielles

Lorsqu'une instance Amazon EC2 d'un cluster qui est basé sur l'AMI Amazon Linux par défaut pour Amazon EMR se lance pour la première fois, elle vérifie les référentiels de package activés pour Amazon Linux et Amazon EMR afin de trouver des mises à jour logicielles qui s'appliquent à la version AMI. Comme avec les autres instances Amazon EC2, les mises à jour critiques et importantes de sécurité provenant de ces référentiels sont automatiquement installées. Pour en savoir plus, consultez Référentiel de package dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. D'autres packages de logiciels et mises à jour de noyau ne sont pas installés, car cela risquerait d'introduire des erreurs de compatibilité.

Lorsque vous vous connectez à une instance de cluster à l'aide de SSH, les premières lignes de sortie d'écran fournissent un lien vers les notes de mise à jour pour l'AMI Amazon Linux qu'utilise l'instance, un avis de la version AMI Amazon Linux la plus récente, un avis sur le nombre de package disponibles pour la mise à jour à partir de référentiels activés, ainsi qu'une directive pour exécuter `sudo yum update`.

> **Important**
>
> Nous vous conseillons fortement de ne pas exécuter `sudo yum update` sur des instances de cluster, que vous soyez connecté à SSH ou que vous utilisez une action d'amorçage. Cela pourrait entraîner des erreurs de compatibilité, car tous les packages sont installés sans distinction.

Bonnes pratiques pour la gestion des mises à jour logicielles

- Si vous utilisez une version antérieure d'Amazon EMR, envisagez et testez une migration à la dernière version avant de mettre à jour les packages logiciels.
- Si vous migrez vers une version ultérieure ou vous mettez à jour vos packages logiciels, testez d'abord son implémentation dans un environnement non-productif. L'option de cloner les cluster à l'aide de la console de gestion Amazon EMR peut vous aider pour ceci.
- Évaluez individuellement les mises à jour logicielles pour vos applications et pour votre version d'AMI Amazon Linux. Testez et installez uniquement les packages les plus nécessaires dans vos environnements de production pour la posture de sécurité, la fonctionnalité d'application ou la performance.
- Surveillez le Centre de sécurité Amazon Linux pour des mises à jour.
- Évitez d'installer des packages en vous connectant à des instances de cluster individuelles à l'aide de SSH. Utilisez plutôt une action d'amorçage pour installer et mettre à jour les packages sur toutes les instances de cluster. Cela nécessite la résiliation d'un cluster et sa réinitialisation. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125).

Utilisation d'une image AMI personnalisée

Lorsque vous utilisez les versions 5.7.0 ou ultérieures d'Amazon EMR, vous pouvez choisir de spécifier une AMI Amazon Linux personnalisée au lieu de l'AMI Amazon Linux par défaut pour Amazon EMR. Une AMI personnalisée est utile si vous souhaitez effectuer les actions suivantes :

- Pré-installez les applications et effectuez d'autres personnalisations au lieu d'utiliser les actions d'amorçage. Cela peut améliorer le temps de lancement du cluster et peut rationaliser le flux de travail de lancement. Pour plus d'informations et pour voir un exemple, consultez Création d'une AMI Amazon Linux personnalisée à partir d'une instance préconfigurée (p. 120).
- Implémentez des configurations de cluster et de nœud plus sophistiquées que les actions d'amorçage ne l'autorise.
- Chiffrez les volumes du périphérique racine EBS (volumes de démarrage) des instances EC2 dans votre cluster si vous utilisez une version de Amazon EMR antérieure à 5.24.0. Pour plus d'informations, veuillez consulter Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré (p. 121).

  Note

  À partir de Amazon EMR version 5.24.0, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous spécifiez AWS KMS comme fournisseur de clé. Pour plus d'informations, consultez Chiffrement de disque local (p. 201).

Bonnes pratiques et considérations

Lorsque vous créez une AMI personnalisée pour Amazon EMR, soyez conscient de ce qui suit :

- Amazon EMR 5.30.0 et la série Amazon EMR 6.x sont basés sur Amazon Linux 2. Pour ces versions EMR, vous devez utiliser des images basées sur Amazon Linux 2 pour les AMI personnalisées. Pour trouver une AMI personnalisée de base, consultez Recherche d'une AMI Linux.
- Pour les versions d'Amazon EMR antérieures aux versions 5.30.0 et 6.x, les AMI Amazon Linux 2 ne sont pas prises en charge. Vous devez utiliser une AMI Amazon Linux 64 bits. Les AMI Amazon Linux avec plusieurs volumes Amazon EBS ne sont pas prises en charge.
- Basez votre personnalisation sur l'AMI Amazon Linux la plus récente, basée sur EBS. Pour consulter la liste des AMI Amazon Linux et des ID d'AMI correspondants, consultez AMI Amazon Linux.
- Ne copiez pas d'instantané d'une instance Amazon EMR existante pour créer une AMI personnalisée. Cela peut entraîner des erreurs.
- Seul le type de virtualisation HVM et les instances compatibles avec Amazon EMR sont pris en charge. Assurez-vous de sélectionner l'image HVM et un type d'instance compatible avec Amazon EMR à mesure que vous suivez le processus de customisation AMI. Pour les types d'instances et de virtualisation compatibles, consultez Types d'instances pris en charge (p. 132).
- Le rôle de votre service doit disposer d'autorisations de lancement sur l'AMI, donc soit l'AMI doit être publique, soit vous devez être le propriétaire de l'AMI, soit elle doit avoir été partagée avec vous par le propriétaire.
- La création d'utilisateurs sur l'AMI avec le même nom que les applications cause des erreurs (par exemple, `hadoop`, `hdfs`, `yarn` ou `spark`).
- Les contenus de `/tmp`, `/var` et `/emr` (s'ils existent sur l'AMI) sont déplacés vers `/mnt/tmp`, `/mnt/var` et `/mnt/emr` respectivement durant le démarrage. Les fichiers sont préservés, mais s'il existe un grand nombre de données,le démarrage peut prendre plus de temps que prévu.
- Si vous utilisez une AMI Linux d'Amazon basé sur une AMI Linux Amazon avec une date de création de 2018-08-11, le serveur Oozie ne démarrera pas. Si vous utilisez Oozie, créez une AMI personnalisée

basée sur un ID d'AMI Linux Amazon avec une date de création différente. Vous pouvez utiliser la commande AWS CLI suivante pour retourner une liste d'ID d'image pour toutes les AMI Linux Amazon HVM avec la version 2018.03, ainsi qu'avec la date de publication, afin que vous puissiez choisir comme base l'AMI Linux Amazon approprié. Remplacez MyRegion par votre identificateur de région, comme us-west-2 par exemple.

```
aws ec2 --region MyRegion describe-images --owner amazon --query 'Images[?Name!=`null`]|
[?starts_with(Name, `amzn-ami-hvm-2018.03`) == `true`].[CreationDate,ImageId,Name]' --
output text | sort -rk1
```

Pour en savoir plus, consultez Création d'une AMI Linux basée sur Amazon EBS dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Spécification d'une AMI personnalisée

Vous pouvez spécifier un ID d'AMI personnalisé lorsque vous créez un cluster en utilisant AWS Management Console, l'AWS CLI, Amazon CloudWatch ou l'API d'Amazon EMR. L'AMI doit exister dans la même région AWS que celle où vous créez le cluster.

Pour spécifier une AMI personnalisée à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Sous Configuration des logiciels, pour Version, choisissez emr-5.7.0 ou une version ultérieure, puis choisissez d'autres options selon votre application. Choisissez Suivant.
4. Sélectionnez des valeurs sous Configuration du matériel appropriées pour votre application, puis choisissez Suivant.
5. Sous Options supplémentaires, pour ID d'AMI personnalisée, entrez une valeur et laissez l'option de mise à jour sélectionnée. Pour en savoir plus sur la modification de cette option de mise à jour, consultez Gestion des mises à jour de référentiel de package d'AMI (p. 120).

6. Pour lancer le cluster, choisissez Suivant, puis remplissez d'autres options de configuration.

Spécification d'une AMI personnalisée à l'aide d'AWS CLI

- Utilisez le `--custom-ami-id` paramètre pour spécifier l'ID d'AMI lorsque vous exécutez la `aws emr create-cluster` commande.

  L'exemple suivant spécifie un cluster qui utilise une AMI personnalisée avec un volume de démarrage de 20 Gio. Pour plus d'informations, consultez Spécification de la taille du volume du périphérique racine Amazon EBS (p. 123).

> **Note**
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "Cluster with My Custom AMI" \
--custom-ami-id MyAmiID --ebs-root-volume-size 20 \
--release-label emr-5.7.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

Gestion des mises à jour de référentiel de package d'AMI

Lors du premier démarrage, par défaut, les AMI Amazon Linux se connectent aux référentiels de package pour installer des mises à jour de sécurité avant le démarrage d'autres services. Selon vos besoins, vous pouvez choisir de désactiver ces mises à jour lorsque vous spécifiez une AMI personnalisée pour Amazon EMR. L'option de désactiver cette fonction est disponible uniquement si vous utilisez une AMI personnalisée.

> **Warning**
>
> Nous vous recommandons fortement de choisir de mettre à jour tous les packages installés sur le démarrage lorsque vous spécifiez une AMI personnalisée. Le choix de ne pas mettre à jour les packages crée un risque sécuritaire supplémentaire.

Si vous utilisez la AWS Management Console, vous pouvez sélectionner l'option de désactiver les mises à jour lorsque vous choisissez ID d'AMI personnalisée.

Si vous utilisez l'AWS CLI, vous pouvez spécifier `--repo-upgrade-on-boot NONE` et `--custom-ami-id` lorsque vous utilisez la commande create-cluster.

Si vous utilisez l'API Amazon EMR, vous pouvez spécifier `NONE` pour le paramètre RepoUpgradeOnBoot.

Création d'une AMI Amazon Linux personnalisée à partir d'une instance préconfigurée

Les étapes basiques de la pré-installation d'un logiciel et de la réalisation d'autres configurations pour créer une AMI Amazon Linux personnalisée pour Amazon EMR sont les suivantes :

- Lancez une instance à partir de l'AMI Amazon Linux de base.
- Connectez-vous à l'instance pour installer le logiciel et réaliser d'autres personnalisations.
- Créez une nouvelle image (instantané d'AMI) de l'instance que vous avez configurée.

Une fois que vous avez créé l'image basée sur votre instance personnalisée, vous pouvez la copier vers une cible chiffrée comme décrit dans la rubrique Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré (p. 121).

Didacticiel : Création d'une AMI à partir d'une instance avec un logiciel personnalisé installé

Pour lancer une instance EC2 basée sur l'AMI Amazon Linux la plus récente

1. Utilisez l'AWS CLI pou exécuter la commande suivante, qui crée une instance à partir d'une AMI existante. Remplacez *MyKeyName* par la paire de clés que vous utilisez pour vous connecter à

l'instance, et *MyAmiId* par l'ID d'une AMI Amazon Linux appropriée. Pour consulter les ID d'AMI les plus récents, voir AMI Amazon Linux.

> **Note**
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws ec2 run-instances --image-id MyAmiID \
--count 1 --instance-type m5.xlarge \
--key-name MyKeyName --region us-west-2
```

La valeur de sortie `InstanceId` est utilisée en tant que *MyInstanceId* dans l'étape suivante.

2. Exécutez la commande suivante :

```
aws ec2 describe-instances --instance-ids MyInstanceId
```

La valeur de sortie `PublicDnsName` est utilisée pour se connecter à l'instance dans l'étape suivante.

Pour se connecter à l'instance et installer le logiciel

1. Utilisez une connexion SSH qui vous permet d'exécuter des commandes shell sur votre instance Linux. Pour plus d'informations, consultez la page Connexion à votre instance Linux à l'aide de SSH dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
2. Effectuez toutes les personnalisations obligatoires. Par exemple :

```
sudo yum install MySoftwarePackage
sudo pip install MySoftwarePackage
```

Pour créer un instantané à partir de votre image personnalisée

- Après avoir personnalisé l'instance, utilisez la commande `create-image` pour créer une AMI à partir de l'instance.

```
aws ec2 create-image --no-dry-run --instance-id MyInstanceId --name MyEmrCustomAmi
```

La valeur de sortie `imageID` est utilisée lorsque vous lancez le cluster ou créez un instantané chiffré. Pour de plus amples informations, veuillez consulter Spécification d'une AMI personnalisée (p. 119) et Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré (p. 121).

Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré

Pour chiffrer le volume du périphérique racine Amazon EBS d'une AMI Amazon Linux pour Amazon EMR, copiez une image d'instantané à partir d'une AMI non chiffrée vers une cible chiffrée. Pour plus d'informations sur la création de volumes EBS, consultez Chiffrement Amazon EBS dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. L'AMI source pour l'instantané peut être l'AMI Amazon Linux de base, ou vous pouvez copier un instantané depuis une AMI dérivée de l'AMI Amazon Linux de base que vous avez personnalisée.

> **Note**
>
> À partir de Amazon EMR version 5.24.0, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous spécifiez AWS KMS comme fournisseur de clé. Pour plus d'informations, consultez Chiffrement de disque local (p. 201).

You can use an external key provider or an AWS customer master key (CMK) to encrypt the EBS root volume. Le rôle de service utilisé par Amazon EMR (habituellement `EMR_DefaultRole`, par défaut) doit être autorisé pour chiffrer et déchiffrer le volume, au minimum, pour qu'Amazon EMR crée un cluster utilisant l'AMI. Lorsque vous utilisez AWS KMS comme fournisseur de clé, cela signifie que les actions suivantes doivent être autorisées :

- `kms:encrypt`
- `kms:decrypt`
- `kms:ReEncrypt*`
- `kms:CreateGrant`
- `kms:GenerateDataKeyWithoutPlaintext"`
- `kms:DescribeKey"`

Le moyen le plus simple pour ce faire est d'ajouter le rôle en tant qu'utilisateur principal, comme décrit dans le didacticiel suivant. La déclaration de stratégie ci-dessous est fournie au cas où vous auriez besoin de personnaliser des stratégies de rôle.

```
{
"Version": "2012-10-17",
"Statement": [
   {
   "Sid": "EmrDiskEncryptionPolicy",
   "Effect": "Allow",
   "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:CreateGrant",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:DescribeKey"
      ],
   "Resource": [
      "*"
      ]
   }
]
}
```

Didacticiel : Création d'une AMI personnalisée avec un volume de périphérique racine chiffré en utilisant un KMS CMK

La première étape de cet exemple est de trouver l'ARN d'un KMS CMK ou d'en créer une nouvelle. Pour plus d'informations sur la création de clés, consultez Création de clés dans le AWS Key Management Service Developer Guide. La procédure suivante vous montre comment ajouter le rôle de service par défaut, `EMR_DefaultRole`, en tant qu'utilisateur de clé, à la stratégie de clé. Notez la valeur ARN de la clé lors de sa création ou de sa modification. Vous utilisez l'ARN plus tard, lorsque vous créez l'AMI.

Ajout du rôle du service pour Amazon EC2 à la liste des utilisateurs de clés de chiffrement en utilisant la console

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Sélectionnez l'alias de la clé CMK à utiliser.
4. Sur la page de détails de la clé, sous Key Users (Utilisateurs de clés), choisissez Add (Ajouter).
5. Dans la boîte de dialogue Attacher, choisissez le rôle de service Amazon EMR. Le nom du rôle par défaut est `EMR_DefaultRole`.
6. Choisissez Attacher.

Création d'une AMI chiffrée en utilisant l'AWS CLI

- Utilisez la commande `aws ec2 copy-image` de l'AWS CLI pour créer une AMI avec un volume de périphérique racine EBS chiffré et la clé que vous avez modifiée. Remplacez la valeur `--kms-key-id` spécifiée, par l'ARN complet de la clé que vous avez précédemment créée ou modifiée.

  > **Note**
  >
  > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws ec2 copy-image --source-image-id MyAmiId \
--source-region us-west-2 --name MyEncryptedEMRAmi \
--encrypted --kms-key-id arn:aws:kms:us-west-2:12345678910:key/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx
```

La sortie de la commande fournit l'ID de l'AMI que vous avez créée, que vous pouvez spécifier lorsque vous créez un cluster. Pour plus d'informations, consultez Spécification d'une AMI personnalisée (p. 119). Vous pouvez également choisir de personnaliser cette AMI en installant un logiciel et en réalisant d'autres configurations. Pour plus d'informations, consultez Création d'une AMI Amazon Linux personnalisée à partir d'une instance préconfigurée (p. 120).

Spécification de la taille du volume du périphérique racine Amazon EBS

Cette option est uniquement disponible avec les versions 4.x ou ultérieures d'Amazon EMR. Vous pouvez spécifier la taille du volume de 10 Gio (par défaut) jusqu'à 100 Gio lorsque vous créez un cluster en utilisant la AWS Management Console, l'AWS CLI ou l'API d'Amazon EMR. Le dimensionnement s'applique uniquement au volume du périphérique racine EBS et à toutes les instances dans le cluster. Il ne s'applique pas aux volumes de stockage, que vous spécifiez séparément pour chaque type d'instance lorsque vous créez votre cluster.

> **Note**
>
> Si vous utilisez l'AMI par défaut, Amazon EMR attache le volume à usage général SSD (GP2) en tant que type de volume du périphérique racine. Une AMI personnalisée peut avoir différents types de volume de périphérique racine. Pour plus d'informations, consultez Spécification d'une AMI personnalisée (p. 119).

Le coût du volume de périphérique racine EBS est calculé au prorata du nombre d'heures en fonction des frais EBS mensuels pour ce type de volume dans la région où s'exécute le cluster. Ceci s'applique

également aux volumes de stockage. Les frais sont exprimés en Go, mais vous spécifiez la taille du volume racine en Gio, vous voudrez peut-être donc prendre en compte ceci dans vos estimations (1 Go correspond à 0.931323 Gio). Pour estimer les frais associés aux volumes du périphérique racine EBS dans votre cluster, utilisez la formule suivante :

($EBS GBmonth)×0.931323÷30÷24×EMR_EBSRootGiB×InstanceCount

Par exemple, prenez un cluster avec un nœud principal et un nœud principal qui utilise l'AMI Amazon Linux de base avec le volume du périphérique racine de 10 Gio par défaut. Si le coût de l'EBS dans la région est de 0,10 USD/Go-Mois, cela correspond à environ 0,00129 USD par instance par heure et à 0,00258 USD par heure pour le cluster (0,10 USD Go-mois divisé par 30 jours, divisé par 24 heures, multiplié par 10 Go, multiplié par 2 instances de cluster).

Pour spécifier la taille du volume du périphérique racine EBS en utilisant la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Sous Configuration des logiciels, pour Version, choisissez une valeur 4.x ou 5.x et autant d'autres options que nécessaire pour votre application et choisissez Suivant.
5. Sous Configuration du matériel, pour taille du volume EBS de périphérique racine, saisissez une valeur entre 10 Gio et 100 Gio.

Pour spécifier la taille du volume du périphérique racine EBS en utilisant l'AWS CLI

- Utilisez le paramètre `--ebs-root-volume-size` de la commande create-cluster, comme illustré dans l'exemple qui suit.

  > **Note**
  >
  > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --release-label emr-5.7.0 \
--ebs-root-volume-size 20 --instance-groups InstanceGroupType=MASTER,\
InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge
```

Configure Cluster Software

Lorsque vous sélectionnez une version d'un logiciel, Amazon EMR utilise une Amazon Machine Image (AMI) avec Amazon Linux pour installer le logiciel que vous choisissez lorsque vous lancez votre cluster, comme Hadoop, Spark et Hive. Amazon EMR fournit régulièrement de nouvelles versions pour ajouter de nouvelles fonctionnalités, de nouvelles applications et des mises à jour générales. Nous vous recommandons d'utiliser la version la plus récente pour lancer votre cluster, chaque fois que possible. La dernière version est l'option par défaut lorsque vous lancez un cluster à partir de la console.

Pour plus d'informations sur les versions d'Amazon EMR et les versions des logiciels disponibles avec cette version, accédez à Amazon EMR Guide de version. Pour plus d'informations sur la manière de modifier les configurations par défaut des applications et des logiciels installés sur votre cluster, consultez Configuration des applications dans le Amazon EMR Guide de version. Certaines versions des composants d'écosystèmes open source Hadoop et Spark qui sont inclus dans les versions Amazon EMR présentent des correctifs et des améliorations, documentés dans Amazon EMR Guide de version.

En plus des logiciels et des applications standard qui peuvent être installés sur votre cluster, vous pouvez utiliser des actions d'amorçage pour installer des logiciels personnalisés. Les actions d'amorçage sont des scripts qui s'exécutent sur les instances lorsque votre cluster est lancé, et qui s'exécutent sur les nouveaux nœuds qui sont ajoutés à votre cluster lorsqu'ils sont créés. Les actions d'amorçage sont également utiles pour appeler des commandes de l'AWS CLI sur chaque nœud afin de copier des objets depuis Amazon S3 dans chaque nœud de votre cluster.

> Note
>
> Les actions d'amorçage sont utilisées différemment dans Amazon EMR version 4.x et les versions ultérieures. Pour plus d'informations sur ces différences par rapport aux versions de Amazon EMR AMI 2.x et 3.x, consultez Différences introduites dans 4.x dans le Amazon EMR Guide de version.

Create Bootstrap Actions to Install Additional Software

Vous pouvez utiliser un bootstrap action pour installer des logiciels supplémentaires ou personnaliser la configuration des instances de cluster. Les actions d'amorçage sont des scripts qui s'exécutent sur le cluster après le lancement de l'instance par Amazon EMR à l'aide de l'AMI (Amazon Machine Image) Linux. Les actions d'amorçage s'exécutent avant qu'Amazon EMR n'installe les applications que vous spécifiez lors de la création du cluster et avant que les nœuds de cluster ne commencent le traitement des données. Si vous ajoutez des nœuds à un cluster en cours d'exécution, des actions d'amorçage s'exécutent également sur ces nœuds de la même façon. Vous pouvez créer des actions amorçage personnalisées et les spécifier quand vous créez votre cluster.

La plupart des actions d'amorçage prédéfinies pour l'AMI Amazon EMR versions 2.x et 3.x ne sont pas prises en charge dans les versions 4.x de Amazon EMR. Par exemple, `configure-Hadoop` et `configure-daemons`) ne sont pas pris en charge dans la version 4.x de Amazon EMR. La version 4.x de Amazon EMR propose plutôt cette fonctionnalité à l'origine. Pour plus d'informations sur la migration des actions d'amorçage des versions AMI Amazon EMR 2.x et 3.x vers la version Amazon EMR 4.x, consultez Différences des versions 4.x d'Amazon EMR dans le Amazon EMR Guide de version.

Rubriques

- Bootstrap Action Basics (p. 126)
- Run If Bootstrap Action (p. 126)
- Shutdown Actions (p. 127)
- Use Custom Bootstrap Actions (p. 127)

Bootstrap Action Basics

Les actions d'amorçage s'exécutent en tant qu'utilisateur Hadoop par défaut. Vous pouvez exécuter une action d'amorçage avec des privilèges racine en utilisant `sudo`.

Toutes les interfaces de gestion Amazon EMR prennent en charge des actions d'amorçage. Vous pouvez spécifier jusqu'à 16 actions d'amorçage par cluster en fournissant plusieurs paramètres `bootstrap-actions` depuis la console, l'AWS CLI ou l'API.

A partir de la console Amazon EMR, vous pouvez en option spécifier une action d'amorçage lors de la création d'un cluster.

Lorsque vous utilisez l'interface de ligne de commande, vous pouvez transmettre des références à des scripts d'action d'amorçage sur Amazon EMR en ajoutant le paramètre `--bootstrap-actions` lorsque vous créez le cluster à l'aide de la commande `create-cluster`. La syntaxe pour un paramètre `--bootstrap-actions` est la suivante :

AWS CLI

```
--bootstrap-actions Path="s3://mybucket/filename",Args=[arg1,arg2]
```

Si l'action d'amorçage renvoie un code d'erreur différent de zéro, Amazon EMR le traite comme un échec et résilie l'instance. Si un trop grand nombre d'instances ne réussissent pas leurs actions d'amorçage, alors Amazon EMR arrête le cluster. Si seules quelques instances échouent, Amazon EMR tente de réaffecter les instances ayant échoués et continue. Utilisez le code d'erreur `lastStateChangeReason` du cluster pour identifier les échecs dus à une action d'amorçage.

Run If Bootstrap Action

Amazon EMR fournit cette action d'amorçage prédéfinie pour exécuter une commande sous certaines conditions lorsqu'une valeur spécifique à l'instance se trouve dans le fichier `instance.json` ou `job-flow.json`. La commande peut faire référence à un fichier dans Amazon S3 qu'Amazon EMR peut télécharger et exécuter.

L'emplacement du script est `s3://elasticmapreduce/bootstrap-actions/run-if`.

L'exemple suivant renvoie la chaîne « en cours d'exécution sur le nœud principal » si le nœud est un maître.

Pour exécuter une commande sous certaines conditions à l'aide de l'AWS CLI

Lors de l'utilisation de l'AWS CLI pour inclure une action d'amorçage, spécifiez `Path` et `Args` en tant que liste séparée par des virgules.

- Pour lancer un cluster avec une action d'amorçage qui exécute sous condition une commande lorsqu'une valeur spécifique à l'instance est trouvée dans le `instance.json` ou `job-flow.json` saisissez la commande suivante et remplacez *`myKey`* avec le nom de votre paire de clés EC2.

  ```
  aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --use-default-roles --ec2-attributes KeyName=myKey --applications Name=Hive --instance-count 1 --instance-type m5.xlarge --bootstrap-actions Path="s3://elasticmapreduce/bootstrap-actions/run-if",Args=["instance.isMaster=true","echo running on master node"]
  ```

  Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

> **Note**
>
> Si vous n'avez pas encore créé le rôle de service Amazon EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Shutdown Actions

Un script d'action d'amorçage peut créer une ou plusieurs actions de fin de tâche en écrivant des scripts dans le répertoire `/mnt/var/lib/instance-controller/public/shutdown-actions/`. Lorsqu'un cluster est arrêté, tous les scripts dans ce répertoire sont exécutés en parallèle. Chaque script doit s'exécuter et s'arrêter dans un délai de 60 secondes.

L'exécution des scripts d'action d'arrêt n'est pas garantie si le nœud s'arrête avec une erreur.

> **Note**
>
> Lorsque vous utilisez les versions 4.0 et ultérieures d'Amazon EMR, vous devez créer manuellement le répertoire `/mnt/var/lib/instance-controller/public/shutdown-actions/` sur le nœud principal. Ce répertoire n'existe pas par défaut. Toutefois, après avoir été créés, les scripts de ce répertoire s'exécutent néanmoins avant l'arrêt. Pour plus d'informations sur la connexion au nœud principal pour créer des répertoires, consultez Connexion au nœud maître à l'aide de SSH (p. 368).

Use Custom Bootstrap Actions

Vous pouvez créer un script personnalisé pour effectuer une action personnalisée d'amorçage. Toutes les interfaces Amazon EMR peuvent faire référence à une action d'amorçage personnalisée.

Table des matières :

- Add Custom Bootstrap Actions Using the AWS CLI or the Amazon EMR CLI (p. 127)
- Add Custom Bootstrap Actions Using the Console (p. 128)
- Use a Custom Bootstrap Action to Copy an Object from Amazon S3 to Each Node (p. 129)

Add Custom Bootstrap Actions Using the AWS CLI or the Amazon EMR CLI

L'exemple suivant utilise un script d'action d'amorçage pour télécharger et extraire une archive TAR compressée depuis Amazon S3. L'exemple de script est stocké à l'adresse suivante : https://elasticmapreduce.s3.amazonaws.com/bootstrap-actions/download.sh.

L'exemple de script ressemble à ce qui suit :

```
#!/bin/bash
set -e
wget -S -T 10 -t 5 http://elasticmapreduce.s3.amazonaws.com/bootstrap-actions/file.tar.gz
mkdir -p /home/hadoop/contents
tar -xzf file.tar.gz -C /home/hadoop/contents
```

Pour créer un cluster avec une action d'amorçage personnalisée à l'aide de l'AWS CLI

Lors de l'utilisation de l'AWS CLI pour inclure une action d'amorçage, spécifiez `Path` et `Args` en tant que liste séparée par des virgules. L'exemple suivant n'utilise pas une liste d'arguments.

- Pour lancer un cluster avec une action d'amorçage personnalisée, tapez la commande suivante, remplacez *myKey* avec le nom de votre paire de clés EC2.
  - Linux, UNIX, and Mac OS X users:

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://elasticmapreduce/bootstrap-actions/download.sh"
```

  - Windows users:

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --use-default-
roles --ec2-attributes KeyName=myKey --applications Name=Hive Name=Pig --instance-
count 3 --instance-type m5.xlarge --bootstrap-actions Path="s3://elasticmapreduce/
bootstrap-actions/download.sh"
```

Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

> **Note**
>
> Si vous n'avez pas encore créé le rôle de service Amazon EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Add Custom Bootstrap Actions Using the Console

La procédure suivante décrit comment utiliser votre propre action d'amorçage personnalisée.

Pour créer un cluster avec une action d'amorçage personnalisée à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Cliquez sur Accéder aux options avancées.
4. Dans Créer cluster -Options avancées, Étapes 1 et 2 choisissez les options souhaitées et passez à Étape 3 : Paramètres généraux du cluster.
5. Sous Bootstrap Actions (Actions d'amorçage) sélectionnez Configure and add (Configurer et ajouter) pour spécifier le nom, l'emplacement du JAR et les arguments pour votre action d'amorçage. Choisissez Add (Ajouter).
6. Vous pouvez en option ajouter des actions d'amorçage comme vous le souhaitez.
7. Procédez à la création du cluster. Vos actions d'amorçage seront effectuées une fois que le cluster a été mis en service et initialisé.

Lorsque le nœud principal du cluster est en cours d'exécution, vous pouvez vous connecter au nœud principal et voir les fichiers journaux que le script d'action d'amorçage a généré dans le répertoire `/mnt/var/log/bootstrap-actions/1`.

Related Topics

- View Log Files (p. 333)

Use a Custom Bootstrap Action to Copy an Object from Amazon S3 to Each Node

Vous pouvez utiliser une action d'amorçage pour copier les objets depuis Amazon S3 vers chaque nœud d'un cluster avant que vos applications ne soient installées. L'AWS CLI est installée sur chaque nœud d'un cluster et, par conséquent, votre action d'amorçage peut appeler les commandes de l'AWS CLI.

L'exemple suivant illustre un simple script d'action d'amorçage qui copie un fichier, `myfile.jar`, depuis Amazon S3 vers un dossier local, `/mnt1/myfolder`, sur chaque nœud de cluster. Le script est enregistré sur Amazon S3 avec le nom de fichier `copymyfile.sh` et les contenus suivants.

```
#!/bin/bash
    aws s3 cp s3://mybucket/myfilefolder/myfile.jar /mnt1/myfolder
```

Lorsque vous lancez le cluster, vous spécifiez le script. L'exemple d'AWS CLI suivant illustre ce scénario :

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.30.1 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://mybucket/myscriptfolder/copymyfile.sh"
```

Configuration du matériel et de la mise en réseau d'un cluster

Lors de la création d'un cluster EMR, il est important de tenir compte de la manière dont vous configurez les instances Amazon EC2 et les options réseau. Les instances EC2 dans un cluster EMR sont organisées en types de nœud. Il existe trois types de nœud : le nœud principal, le nœud de noyau et les nœuds de tâche. Chaque type de nœud exécute un ensemble de rôles définis par les applications distribuées que vous installez sur le cluster. Au cours d'une tâche Hadoop MapReduce ou Spark, par exemple, les composants des nœuds principaux et des nœuds de tâche traitent les données, transfèrent la sortie vers Amazon S3 ou HDFS, et fournissent les métadonnées de statut en retour au nœud principal. Dans le cas d'un cluster à un seul nœud, tous les composants s'exécutent sur le nœud principal.

La collection d'instances EC2 qui héberge chaque type de nœud est appelée un parc d'instances ou un groupe d'instances uniforme. La configuration des parcs d'instances ou des groupes d'instances uniformes est un choix que vous effectuez lors de la création d'un cluster. Elle s'applique à tous les types de nœud et elle ne peut pas être modifiée ultérieurement.

Lorsque vous créez un cluster, vous faites des choix qui déterminent au final le profil de performance de votre cluster. Ce chapitre traite ces options en détail et les rassemble dans les bonnes pratiques et les consignes.

> Note
>
> La configuration des parcs d'instances est disponible uniquement dans les versions 4.8.0 et ultérieures d'Amazon EMR, à l'exception des versions 5.0.0 et 5.0.3.

Rubriques

- Comprendre les nœuds principaux, de noyau et de tâche (p. 130)
- Configurer les instances EC2 (p. 131)
- Configuration de la mise en réseau (p. 137)
- Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes (p. 148)
- Consignes et bonnes pratiques pour la configuration de cluster (p. 161)

Comprendre les nœuds principaux, de noyau et de tâche

Utilisez cette section pour comprendre comment Amazon EMR utilise chacun de ces types de nœuds comme base pour la planification de capacité de cluster.

Nœud maître

Le nœud principal gère le cluster et exécute généralement les composants maîtres des applications distribuées. Par exemple, le nœud principal exécute le service YARN ResourceManager pour gérer les ressources des applications, ainsi que le service HDFS NameNode. Il suit également le statut des tâches soumises au cluster et surveille l'intégrité des groupes d'instances.

Pour surveiller la progression d'un cluster et interagir directement avec les applications, vous pouvez vous connecter au nœud principal via SSH en tant qu'utilisateur Hadoop. Pour plus d'informations, consultez la section Connexion au nœud maître à l'aide de SSH (p. 368). La connexion au nœud principal vous permet d'accéder directement aux répertoires et aux fichiers, tels que les fichiers journaux Hadoop. Pour plus d'informations, consultez la section View Log Files (p. 333). Vous pouvez aussi afficher les interfaces utilisateur que les applications publient sous forme de sites web s'exécutant sur le nœud maître. Pour plus d'informations, consultez la section Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

> Note
>
> Avec Amazon EMR version 5.23.0 ou ultérieure, vous pouvez lancer un cluster doté de trois nœuds principaux afin de prendre en charge la haute disponibilité des applications telles que YARN Resource Manager, HDFS Name Node, Spark, Hive et Ganglia. Le nœud principal n'est plus un point de défaillance potentiel grâce à cette fonctionnalité. Si l’un des nœuds principaux tombe en panne, Amazon EMR passe automatiquement sur un nœud principal de secours et remplace le nœud principal défaillant par un nouveau nœud ayant la même configuration et les mêmes actions de démarrage. Pour plus d’informations, consultez Planification et configuration de nœuds principaux.

Nœuds principaux

Les nœuds principaux sont gérés par le nœud maître. Les nœuds principaux exécutent le démon de nœud de données pour coordonner le stockage des données dans le cadre du système de fichiers distribué Hadoop (HDFS). Ils exécutent également le démon du dispositif de suivi des tâches et exécutent d'autres tâches de calcul parallèles sur les données dont ont besoin les applications installées. Par exemple, un nœud principal exécute les démons YARN NodeManager, les tâches Hadoop MapReduce et les exécuteurs Spark.

Toutefois, à la différence du nœud principal, il peut y avoir plusieurs nœuds principaux, et par conséquent plusieurs instances EC2, dans le groupe d'instances ou le parc d'instances. Il y a un seul groupe d'instances ou parc d'instances principal. Avec les groupes d'instances, vous pouvez ajouter et retirer des instances EC2 tandis que le cluster est en cours d'exécution ou définir un dimensionnement automatique. Pour plus d'informations sur l'ajout et le retrait d'instances EC2 avec la configuration des groupes d'instance, consultez Dimensionnement des ressources de cluster (p. 384). Avec les parcs d'instances, vous pouvez ajouter et retirer efficacement des instances en modifiant les capacités cibles du parc

d'instances pour les instances à la demande et Spot, comme il convient. Pour plus d'informations sur les capacités cibles, consultez Options de parc d'instances (p. 150).

> Warning
>
> La suppression des démons HDFS à partir d'un nœud de noyau en cours d'exécution ou la suppression de nœuds de noyau peuvent engendrer une perte de données. Faites attention lorsque vous configurez des nœuds de noyau sur des instances Spot. Pour plus d'informations, consultez la section Quand faut-il utiliser des instances Spot ? (p. 162).

Nœuds de tâches

Les nœuds de tâches sont facultatifs. Vous pouvez les utiliser afin d'ajouter de la puissance pour l'exécution de tâches de calcul parallèles, comme les tâches Hadoop MapReduce et les exécuteurs Spark. Les nœuds de tâches n'exécutent pas le démon de nœud de données et ne stockent pas les données dans HDFS. Comme avec les nœuds principaux, vous pouvez ajouter des nœuds de tâches à un cluster en ajoutant des instances EC2 à un groupe d'instances existant, ou en modifiant les capacités cibles d'un parc d'instances de tâches. Les clusters avec la configuration de groupe d'instances uniforme peuvent comporter un total de 48 groupes d'instances de tâches. La possibilité d'ajouter des groupes d'instances uniformes de cette manière vous permet de combiner des types d'instance EC2et des options de tarification, comme les instances à la demande et les instances Spot. Vous pouvez ainsi répondre aux exigences de charge de travail de manière rentable. Lorsque vous utilisez la configuration de parc d'instances pour votre cluster, la possibilité de combiner des types d'instance et des options d'achat est intégrée, de sorte qu'il n'y a qu'un seul parc d'instances de tâches.

Les instances Spot étant souvent utilisées pour exécuter des nœuds de tâche, Amazon EMR dispose d'une fonctionnalité par défaut pour la planification de tâches YARN de sorte que les tâches en cours d'exécution n'échouent pas lorsque des nœuds de tâche en cours d'exécution sur des instances Spot sont résiliées. Amazon EMR s'en assure en n'autorisant les processus maîtres des applications à s'exécuter uniquement sur des nœuds principaux. Le processus maître de l'application contrôle l'exécution des tâches et doit rester actif pendant toute la durée de vie de la tâche.

Amazon EMR version 5.19.0 et ultérieures utilise la fonction des étiquettes de nœud YARN intégrés pour y parvenir. (Les versions antérieures utilisaient un correctif de code). Les propriétés des classifications de configuration `yarn-site` et `capacity-scheduler` sont configurées par défaut de sorte que le planificateur de YARN et le planificateur d'équité tirent parti des étiquettes de nœud. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de sorte que les masters de l'application ne soient planifiés que sur des nœud avec l'étiquette CORE. La modification manuelle de propriétés apparentées dans les classifications de configuration yarn-site et capacity-scheduler ou directement dans les fichiers XML associés pourrait rompre cette fonctionnalité ou la modifier.

À partir de la série Amazon EMR version 6.x, la fonction des étiquettes de nœud YARN est désactivée par défaut. Les processus principaux des applications peuvent s'exécuter à la fois sur les nœuds de noyau et sur les nœuds de tâche par défaut. Vous pouvez activer la fonction d'étiquetage des nœuds YARN en configurant les propriétés suivantes :

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Pour plus d'informations sur les propriétés spécifiques, consultez Paramètres Amazon EMR pour empêcher l'échec de tâche en raison d'une résiliation d'instance Spot de nœud de tâche (p. 163).

Configurer les instances EC2

Les instances EC2 se déclinent dans différentes configurations, également appelées types d'instance. Chaque type d'instance offre des capacités d'UC, d'entrée/sortie et de stockage différentes. En plus du type d'instance, vous pouvez choisir différentes options d'achat pour les instances EC2. Vous pouvez spécifier

différents types d'instance et options d'achat au sein des groupes d'instances ou des parcs d'instances. Pour plus d'informations, consultez la section Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes (p. 148). Pour obtenir des conseils sur le choix des options appropriées, consultez Consignes et bonnes pratiques pour la configuration de cluster (p. 161).

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide d'AWS Management Console, le nombre de processeurs virtuels affiché pour chaque type d'instance correspond au nombre de cœurs virtuels YARN pour ce type d'instance, et non au nombre de processeurs virtuels EC2 pour ce type d'instance. Pour de plus amples informations sur le nombre de processeurs virtuels pour chaque type d'instance, veuillez consulter Types d'instances Amazon EC2.

Rubriques

- Types d'instances pris en charge (p. 132)
- Options d'achat d'instance (p. 133)
- Stockage d'instance (p. 136)

Types d'instances pris en charge

Le tableau suivant décrit les types d'instance pris en charge par Amazon EMR. Pour plus d'informations, consultez Instances Amazon EC2 et Matrice de type d'instance AMI Amazon Linux.

Tous les types d'instance ne sont pas toutes disponibles dans toutes les régions. Si vous créez un cluster en utilisant un type d'instance qui n'est pas disponible, votre cluster 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, cliquez sur le lien correspondant à l'option d'achat de votre instance et filtrez par Région pour voir si le type d'instance que vous sélectionnez dans la liste ci-dessous est disponible dans la région.

À partir de la version Amazon EMR 5.13.0, toutes les instances utilisent la virtualisation HVM et le stockage basé sur des volumes EBS pour les volumes racine. Lorsque vous utilisez des versions Amazon EMR antérieures à 5.13.0, certaines instances de la génération précédente utilisent la virtualisation PVM. Elles sont indiquées dans le tableau. Pour plus d'informations, consultez la page Types de virtualisations AMI Linux.

Certains types d'instance prennent en charge la mise en réseau améliorée. Pour plus d'informations, consultez Mise en réseau améliorée sous Linux.

Amazon EMR prend en charge `Previous Generation Instances` pour prendre en charge les applications optimisées pour ces instances et qui n'ont pas encore été mises à niveau. Pour plus d'informations sur ces types d'instance et sur les chemins de mise à niveau, consultez Instances de la génération précédente.

Classe d'instance	Types d'instance
Usage général	`m1.small`[1] | `m1.medium`[1] | `m1.large`[1] | `m1.xlarge`[1] | `m2.xlarge`[1] | `m2.2xlarge`[1] | `m2.4xlarge`[1] | `m3.xlarge`[1] | `m3.2xlarge`[1] | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5xlarge | m5a | m5xlarge | m5a | **Note** Les instances m5a-series sont disponibles avec Amazon EMR version 5.20.0 ou ultérieure.

Classe d'instance	Types d'instance
Calcul optimisé	`c1.medium`[1] [2] | `c1.xlarge`[1] | `c3.xlarge`[1] | `c3.2xlarge`[1] | `c3.4xlarge`[1] | `c3.8xlarge`[1] | c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c5x | c5x | c5x |5large | c5x | c5 `cc2.8xlarge` | z1d.xlarge | z1d.2xlarge | z1d.3xlarge | z1d.6xlarge | z1d.12xlarge Note Les instances c5n-series sont disponibles avec Amazon EMR version 5.20.0 ou ultérieure.
Mémoire optimisée	`r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge` | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5xlarge | r5. `cr1.8xlarge` Note Les instances r5a-series sont disponibles avec Amazon EMR version 5.20.0 ou ultérieure.
Stockage optimisé	h1.2xlarge | h1.4xlarge | h1.8xlarge | h1.16xlarge | `hs1.8xlarge`[1] | `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge` | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge | i3en.xlarge | i3en.2xlarge |i3en.3xlarge |i3en.6xlarge |i3en.12xlarge |i3en.24xlarge |d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge Note Les instances i3-series sont disponibles avec Amazon EMR version 5.9.0 ou ultérieure. Les instances i3en-series sont disponibles avec Amazon EMR version 5.25.0 ou ultérieure.
Instances GPU	`g2.2xlarge` | g3.4xlarge | g3.8xlarge | g3.16xlarge | g3s.xlarge | g4dn.xlarge | g4dn.2xlarge | g4dn.4xlarge | g4dn.8xlarge | g4dn.12xlarge | g4dn.16xlarge | p2.xlarge | p2.8xlarge | p4xlarge | p3.2xlarge | Note Les pilotes NVIDIA et CUDA sont installés sur les types d'instances P2 et P3 par défaut.

1Utilise l'AMI de virtualisation PVM avec Amazon EMR versions antérieures à 5.13.0. Pour plus d'informations, consultez Types de virtualisation d'AMI Linux.

[2]Non pris en charge dans la version 5.15.0.

[2]Pris en charge dans les versions 5.13.0 et ultérieures.

Options d'achat d'instance

Lorsque vous configurez un cluster, vous choisissez une option d'achat pour les instances EC2. Vous pouvez choisir d'utiliser les instances À la demande et/ou les instances Spot. Les prix varient en fonction du type d'instance et de la région. Pour connaître la tarification actuelle, consultez Tarification Amazon EMR.

Votre choix concernant l'utilisation de groupes d'instances ou de parcs d'instances dans votre cluster détermine la façon dont vous pouvez modifier les options d'achat d'instance lorsqu'un cluster est en

cours d'exécution. Si vous choisissez des groupes d'instances uniformes, le type d'instance et l'option d'achat s'appliquent à toutes les instances EC2 de chaque groupe d'instances, et vous pouvez uniquement spécifier l'option d'achat pour un groupe d'instances lors de sa création. Si vous choisissez des parc d'instances, vous pouvez modifier les options d'achat après la création d'un parc d'instances, et vous pouvez combiner des options d'achat afin de remplir une capacité cible que vous spécifiez. Pour plus d'informations sur ces configurations, consultez Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes (p. 148).

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide d'AWS Management Console, le nombre de processeurs virtuels affiché pour chaque type d'instance correspond au nombre de cœurs virtuels YARN pour ce type d'instance, et non au nombre de processeurs virtuels EC2 pour ce type d'instance. Pour de plus amples informations sur le nombre de processeurs virtuels pour chaque type d'instance, veuillez consulter Types d'instances Amazon EC2.

Instances à la demande

Les instances à la demande vous permettent de payer la capacité de calcul à l'heure. Vous pouvez aussi faire en sorte que ces instances à la demande utilisent les options d'achat de l'instance réservée ou de l'instance dédiée. Les instances réservées vous permettent d'effectuer un seul paiement pour une instance afin de réserver de la capacité. Les instances dédiées sont physiquement isolées au niveau matériel hôte des instances qui appartiennent à d'autres comptes AWS. Pour plus d'informations sur les options d'achat, consultez Options d'achat d'instance dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Utilisation d'instances réservées

Pour utiliser les instances réservées dans Amazon EMR, vous utilisez Amazon EC2 pour acheter l'instance réservée et spécifiez les paramètres de la réservation, y compris l'étendue de la réservation dès lors qu'elle s'applique à une région ou une zone de disponibilité. Pour de plus amples informations, veuillez consulter Instances réservées Amazon EC2 et Achat d'instances réservées dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Une fois que vous avez acheté une instance réservée, si l'ensemble des conditions suivantes sont vérifiées, Amazon EMR utilise l'instance réservée lors du lancement d'un cluster :

- Une instance à la demande est spécifiée dans la configuration de cluster qui correspond à la spécification d'instance réservée
- Le cluster est lancée dans l'étendue de la réservation d'instance (zone de disponibilité ou région)
- La capacité de l'instance réservée est encore disponible

Par exemple, supposons que vous achetez une instance réservée `m5.xlarge` avec la réservation d'instance dont l'étendue est limitée à la région USA Est. Vous lancez ensuite un cluster EMR dans la région USA Est qui utilise deux instances `m5.xlarge`. La première instance est facturée au tarif de l'instance réservée et l'autre est facturée au tarif à la demande. La capacité de l'instance réservée est utilisée avant la création des instances à la demande.

Utilisation d'instances dédiées

Pour utiliser des instances dédiées, vous achetez des instances dédiées avec Amazon EC2, puis vous créez un VPC avec l'attribut de location Dédié. Au sein d'Amazon EMR, vous indiquez ensuite qu'un cluster doit se lancer dans ce VPC. Toute instance à la demande dans le cluster qui correspond à la spécification d'instance dédiée utilise les instances dédiées disponibles lors du lancement du cluster.

> **Note**
>
> Amazon EMR ne prend pas en charge la définition de l'attribut `dedicated` sur des ressources individuelles.

Instances Spot

Les instances Spot dans Amazon EMR fournissent une option qui vous permet d'acheter la capacité d'instance Amazon EC2 à un coût réduit par rapport à un achat à la demande. L'inconvénient d'utiliser des instances Spot est que les instances peuvent se terminer de manière imprévisible si les prix varient. Pour plus d'informations sur l'utilisation appropriée des instances Spot pour votre application, consultez Quand faut-il utiliser des instances Spot ? (p. 162).

Lorsqu'Amazon EC2 a une capacité inutilisée, il propose des instances EC2 à un coût réduit, appelé prix Spot. Ce prix varie en fonction de la disponibilité et de la demande, et il est défini en fonction de la région et de la zone de disponibilité. Lorsque vous choisissez des instances Spot, vous spécifiez le prix Spot que vous êtes prêt à payer pour chaque type d'instance EC2. Lorsque le prix Spot dans la zone de disponibilité du cluster est inférieur au prix Spot maximum spécifié pour ce type d'instance, les instances se lancent. Lors de l'exécution des instances, vous êtes facturé au prix Spot actuel et non au prix Spot maximum.

Lorsque vous créez un cluster avec des parcs d'instances, vous avez la possibilité d'utiliser une durée définie (également appelée bloc d'instances Spot) qui offre un plus haut degré de prévisibilité. Les instances Spot sont résiliées à la fin de cette durée, mais elles ne sont pas interrompues avant l'expiration de celle-ci. Cette rubrique décrit le fonctionnement des instances Spot avec Amazon EMR.

Pour obtenir des informations sur la tarification actuelle, consultez Tarification des instances Spot Amazon EC2. Pour de plus amples informations, veuillez consulter Instances Spot dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Lorsque vous créez et configurez un cluster, vous indiquez les options de réseau qui déterminent au final la zone de disponibilité où se lance votre cluster. Pour plus d'informations, consultez la section Configuration de la mise en réseau (p. 137).

> Tip
>
> Vous pouvez voir le prix Spot en temps réel dans la console lorsque vous survolez l'info-bulle d'informations à côté de l'option d'achat Spot lorsque vous créez un cluster à l'aide des Options avancées. Les prix pour chaque zone de disponibilité dans la zone sélectionnée sont affichés. Les prix les plus bas figurent sur les lignes de couleur verte. En raison des fluctuations des prix Spot entre les zones de disponibilité, la sélection de la zone de disponibilité avec le prix initial le plus bas peut ne pas correspondre au prix le plus bas sur toute la durée de vie du cluster. Pour des résultats optimaux, étudiez l'historique de la tarification des zones de disponibilité avant de choisir. Pour plus d'informations, consultez Historique de tarification d'instance Spot dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Les options d'instance Spot varient selon que vous utilisez des groupes d'instances uniformes ou des parcs d'instances dans votre configuration de cluster.

Instances Spot dans les groupes d'instances uniformes

Lorsque vous utilisez des instances Spot dans un groupe d'instances uniforme, toutes les instances d'un groupe d'instances doivent être des instances Spot. Vous spécifiez un seul sous-réseau ou zone de disponibilité pour le cluster. Pour chaque groupe d'instances, vous spécifiez un seul type d'instance Spot et un prix Spot maximum. Les instances Spot de ce type se lancent si le prix Spot dans la région et la zone de disponibilité du cluster est inférieur au prix Spot maximum. Les instances sont résiliées si le prix Spot dépasse votre prix Spot maximum. Vous définissez le prix Spot maximum uniquement lorsque vous configurez un groupe d'instances. Il ne peut pas être modifié ultérieurement. Pour plus d'informations, consultez la section Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes (p. 148).

Instances Spot dans les parcs d'instances

Lorsque vous utilisez la configuration des parcs d'instances, des options supplémentaires vous offrent davantage de contrôle sur le mode de lancement et de suspension des instances Spot. Fondamentalement, les parcs d'instances utilisent une méthode différente de celle des groupes d'instances

uniformes pour lancer des instances. Vous établissez une capacité cible pour les instances Spot (et les instances à la demande) et jusqu'à cinq types d'instance. Vous pouvez aussi spécifier une capacité pondérée pour chaque type d'instance ou utiliser le vCPU (cœurs virtuels YARN) du type d'instance comme capacité pondérée. Cette capacité pondérée compte pour votre capacité cible lorsqu'une instance de ce type est allouée. Amazon EMR alloue les instances avec les deux options d'achat jusqu'à ce que la capacité cible de chaque cible est atteinte. En outre, vous pouvez définir une plage de zones de disponibilité à partir de laquelle Amazon EMR peut choisir lors du lacement des instances. Vous indiquez également des options d'instance Spot supplémentaires pour chaque parc, y compris un délai de mise en service et, éventuellement, une durée définie. Pour plus d'informations, consultez la section Configuration de parcs d'instances (p. 149).

Stockage d'instance

Le stockage d'instance ou de volume EBS est utilisé pour les données HDFS, ainsi que les tampons, les caches, les données de travail et autres contenus temporaires que certaines applications peuvent « déverser » sur le système de fichiers local. EMRFS peut garantir qu'il existe une « source de vérité » permanente pour les données HDFS stockées dans Amazon S3.

Amazon EBS fonctionne différemment avec Amazon EMR qu'avez les instances Amazon EC2 ordinaires. Les volumes Amazon EBS attachés aux clusters EMR sont éphémères : les volumes sont supprimés à l'arrêt du cluster et des instances (par exemple, lors de la réduction des groupes d'instances). Il est donc important de ne pas compter sur la persistance des données. Les données sont éphémères sur ces volumes, mais il est possible que les données dans HDFS soient répliquées selon le nombre et la spécialisation des nœuds du cluster. Lorsque vous ajoutez des volumes de stockage EBS, ils sont montés en tant que volumes supplémentaires. Ils ne font pas partie du volume racine. YARN est configuré pour utiliser tous les volumes supplémentaires, mais vous êtes responsable de l'allocation des volumes supplémentaires en tant que stockage local (comme pour les fichiers journaux locaux).

Autres mises en garde pour l'utilisation d'Amazon EBS avec les clusters EMR :

- Vous ne pouvez pas créer un instantané d'un volume EBS, puis le restaurer dans Amazon EMR. Pour créer des configurations personnalisées réutilisables, choisissez une AMI personnalisée (disponible dans Amazon EMR version 5.7.0 et ultérieure). Pour plus d'informations, consultez la section Utilisation d'une image AMI personnalisée (p. 118).
- Un volume de périphérique racine EBS chiffré n'est pris en charge que lors de l'utilisation d'une AMI personnalisée. Pour plus d'informations, consultez la section Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré (p. 121). Les volumes de stockage EBS chiffrés ne sont pas pris en charge.
- Si vous appliquez des balises à l'aide de l'API Amazon EMR, ces opérations sont appliquées aux volumes EBS.
- Il y a une limite de 25 volumes par instance.
- Les volumes EBS sur les nœuds principaux ne peuvent pas être inférieurs à 5 Go.

Stockage EBS par défaut pour les instances

Amazon EMR attache automatiquement un disque SSD Usage général (gp2) Amazon EBS 10 Go en tant que périphérique racine pour ses images AMI, afin d'améliorer les performances. En outre, pour les instances EC2 avec un stockage EBS uniquement, Amazon EMR alloue des volumes de stockage EBS aux instances. Lorsque vous créez un cluster à l'aide d'Amazon EMR version 5.22.0 et versions ultérieures, la quantité de stockage EBS par défaut augmente en fonction de la taille de l'instance. En outre, nous avons fractionné une plus grande quantité de stockage sur plusieurs volumes, afin d'offrir de meilleures performances en termes d'IOPS et pour certaines charges de travail normalisées. Si vous voulez utiliser une autre configuration de stockage d'instance EBS, vous pouvez la spécifier lorsque vous créez un cluster EMR ou ajoutez des nœuds à un cluster existant. Consultez le tableau ci-dessous pour identifier le nombre de volumes de stockage EBS par défaut, leur taille et la taille totale par type d'instance.

Les coûts EBS sont calculés au prorata du nombre d'heures en fonction des frais Amazon EBS mensuels pour les volumes gp2 de la région où s'exécute le cluster. Par exemple, le coût EBS par heure pour le volume racine sur chaque nœud de cluster dans une région qui facture 0,10 $/Go/mois serait d'environ 0,00139 $ par heure (0,10 $/Go/mois divisé par 30 jours et divisé par 24 h fois 10 Go).

Volumes de stockage EBS par défaut et taille par type d'instance pour Amazon EMR 5.22.0 et versions ultérieures

Taille d'instance	Nombre de volumes	Taille du volume (Gio)	Total Taille (Gio)
*.large	1	32	32
*.xlarge	2	32	64
*.2xlarge	4	32	128
*.4xlarge	4	64	256
*.8xlarge	4	128	512
*.9xlarge	4	144	576
*.10xlarge	4	160	640
*.12xlarge	4	192	768
*.16xlarge	4	256	1 024
*.18xlarge	4	288	1 152
*.24xlarge	4	384	1 536

Spécification de volumes de stockage EBS supplémentaires

Lorsque vous configurez les types d'instance dans Amazon EMR, vous pouvez spécifier des volumes EBS supplémentaires, ce qui ajoute de la capacité au-delà du stockage d'instance (le cas échéant) et du volume EBS par défaut. Amazon EBS fournit les types de volume suivants : Usage général (SSD), IOPS provisionnés (SSD), débit optimisé (HDD), froid (HDD) et magnétique. Ils se distinguent par leurs caractéristiques de performances et leurs prix, ce qui vous permet d'adapter votre stockage en fonction des besoins d'analyse et d'entreprise de vos applications. Par exemple, certaines applications peuvent avoir besoin de se déverser sur le disque, tandis que d'autres peuvent travailler en toute sécurité dans la mémoire ou à l'aide d'Amazon S3.

Vous pouvez uniquement attacher les volumes EBS aux instances au démarrage du cluster, à moins que vous ajoutiez un groupe d'instances de nœud de tâches supplémentaire, auquel cas vous pouvez ajouter des volumes EBS à ce moment. En cas de défaillance d'une instance dans un cluster EMR, l'instance et les volumes EBS attachés sont remplacés. Par conséquent, si vous détachez manuellement un volume EBS, Amazon EMR traite cela comme une défaillance et remplace le stockage d'instance (le cas échéant) et les stockages de volume.

Configuration de la mise en réseau

Il peut y avoir deux options de plateforme réseau que vous pouvez choisir pour votre cluster : EC2-Classique ou 2e EC-PCV. Dans EC2-Classic, vos instances s'exécutent dans un réseau plat unique partagé avec d'autres clients. EC2-Classic est disponible uniquement avec certains comptes dans certaines régions. Pour de plus amples informations, consultez Amazon EC2 et Amazon VPC dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Dans EC2-VPC, votre cluster utilise Amazon Virtual Private Cloud (Amazon VPC) et les instances EC2 exécutent un VPC qui est logiquement isolé dans

votre compte AWS. Amazon VPC vous permet de provisionner un virtual private cloud (VPC), une zone isolée dans AWS où vous pouvez configurer un réseau virtuel, contrôler certains aspects comme les plages d'adresses IP privées, les sous-réseaux, les tables de routage et les passerelles de réseau.

&VPC propose les fonctions suivantes :

- Traitement des données sensibles

  Le lancement d'un cluster dans un VPC est similaire au lancement d'un cluster dans un réseau privé, avec des outils supplémentaires, tels que des tables de routage et des listes ACL réseau, afin de définir les personnes autorisées à accéder au réseau. Si vous traitez des données sensibles dans votre cluster, vous pouvez profiter du meilleur contrôle des accès que procure le lancement de votre cluster dans un VPC. En outre, vous pouvez choisir de lancer vos ressources dans un sous-réseau privé, dans lequel aucune de ces ressources ne dispose d'une connectivité Internet directe.
- Accès aux ressources sur un réseau interne

  Si votre source de données se trouve dans un réseau privé, le chargement de ces données dans AWS en vue de les importer dans Amazon EMR peut s'avérer impossible ou non recommandé, en raison de la quantité de données à transférer ou du caractère sensible des données. Au lieu de cela, vous pouvez lancer le cluster dans un VPC et connecter votre centre de données à votre VPC via une connexion VPN, ce qui permet au cluster d'accéder aux ressources sur votre réseau interne. Par exemple, si vous avez une base de données Oracle dans votre centre de données, le lancement de votre cluster dans un VPC connecté à ce réseau par VPN permet au cluster d'accéder à la base de données Oracle.

Sous-réseaux publics et privés

Vous pouvez lancer des clusters EMR dans des sous-réseaux VPC publics et privés. Cela signifie que vous n'avez pas besoin de connectivité Internet pour exécuter un cluster EMR. Cependant, vous devrez peut-être configurer la traduction d'adresses réseau (NAT) et les passerelles VPN pour accéder aux services ou aux ressources situés en dehors du VPC, par exemple dans un intranet d'entreprise ou dans des points de terminaison de services AWS publics comme AWS Key Management Service.

> **Important**
>
> Amazon EMR prend uniquement en charge le lancement de clusters dans des sous-réseaux privés à partir de la version 4.2.

Pour de plus amples informations sur Amazon VPC, veuillez consulter Amazon VPC Guide de l'utilisateur.

Rubriques

- Options Amazon VPC (p. 138)
- Configuration d'un VPC pour héberger des clusters (p. 143)
- Lancement de clusters dans un VPC (p. 145)
- Stratégie Amazon S3 minimum pour un sous-réseau privé (p. 147)
- Ressources supplémentaires pour en savoir plus sur les VPC (p. 147)

Options Amazon VPC

Lorsque vous lancez un cluster Amazon EMR au sein d'un VPC, vous pouvez le lancer dans un sous-réseau public, privé ou partagé. Les différences de configuration sont légères mais importantes, en fonction du type de sous-réseau que vous choisissez pour un cluster.

Sous-réseaux publics

Les clusters EMR dans un sous-réseau public nécessitent une passerelle Internet connectée Cela est dû au fait que les clusters Amazon EMR doivent accéder aux services AWS et à Amazon EMR. Si un service,

tel que Amazon S3, offre la possibilité de créer un point de terminaison de VPC, vous pouvez accéder à ces services à l'aide du point de terminaison au lieu d'accéder à un point de terminaison public via une passerelle Internet. En outre, Amazon EMR ne peut pas communiquer avec des clusters dans des sous-réseaux publics via un périphérique de traduction d'adresses réseau (NAT). C'est pour cette raison qu'une passerelle Internet est obligatoire, mais vous pouvez toujours utiliser une instance NAT ou une passerelle pour le reste du trafic dans les scénarios plus complexes.

Toutes les instances d'un cluster se connectent à Amazon S3 via un point de terminaison de VPC ou une passerelle Internet. Les autres services AWS qui actuellement ne prennent pas en charge les points de terminaison de VPC utilisent uniquement une passerelle Internet.

Si vous disposez de ressources AWS supplémentaires que vous ne souhaitez pas connecter à la passerelle Internet, vous pouvez les lancer dans un sous-réseau privé que vous créez dans votre VPC.

Les clusters exécutant un sous-réseau public utilisent deux groupes de sécurité : un groupe pour le nœud principal et un autre pour les nœuds de noyau et de tâche. Pour plus d'informations, consultez la section Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Le schéma suivant montre l'exécution d'un cluster Amazon EMR dans un VPC à l'aide d'un sous-réseau public. Le cluster est capable de se connecter à d'autres ressources AWS, par exemple aux compartiments Amazon S3, via la passerelle Internet.

Le schéma suivant montre comment configurer un VPC afin qu'un cluster présent dans le VPC puisse accéder aux ressources de votre propre réseau, par exemple une base de données Oracle.

Sous-réseaux privés

Les sous-réseaux privés vous permettent de lancer des ressources AWS sans qu'une passerelle Internet ne soit nécessairement attachée au sous-réseau. Cela peut être utile, par exemple, dans une application qui utilise ces ressources privées sur le backend. Ces ressources peuvent ensuite initier le trafic sortant à l'aide d'une instance NAT située dans un autre sous-réseau auquel est attachée une passerelle Internet. Pour plus d'informations sur ce scénario, consultez Scénario 2 : VPC avec sous-réseaux publics et privés (NAT).

> **Important**
>
> Amazon EMR prend uniquement en charge le lancement de clusters dans des sous-réseaux privés à partir de la version 4.2.

Différences avec les sous-réseaux publics :

- Pour accéder aux services AWS qui n'offrent pas de point de terminaison de VPC, vous devez toujours utiliser une instance NAT ou une passerelle Internet.
- Au minimum, vous devez indiquer un chemin vers le compartiment des journaux du service Amazon EMR et vers le répertoire Amazon Linux dans Amazon S3. Pour plus d'informations, veuillez consulter Stratégie Amazon S3 minimum pour un sous-réseau privé (p. 147)
- Si vous utilisez des fonctionnalités EMRFS, vous devez disposer d'un point de terminaison de VPC Amazon S3 et d'un chemin de votre sous-réseau privé vers DynamoDB.
- Le débogage fonctionne uniquement si vous fournissez une route de votre sous-réseau privé vers un point de terminaison Amazon SQS public.
- La création d'une configuration de sous-réseau privé avec une passerelle ou une instance NAT dans un sous-réseau public est uniquement prise en charge à l'aide d'AWS Management Console. Le moyen le plus simple d'ajouter et de configurer des instances NAT et des points de terminaison d'un EMR Amazon S3 pour les clusters EMR consiste à utiliser la page VPC Subnets List (Liste de sous-réseaux VPC) dans la console Amazon EMR. Pour configurer les passerelles NAT, consultez Passerelles NAT dans le Amazon VPC Guide de l'utilisateur.
- Vous ne pouvez pas modifier un sous-réseau avec un cluster EMR existant de public à privé ou inversement. Pour placer un cluster EMR au sein d'un sous-réseau privé, le cluster doit être démarré dans ce sous-réseau privé.

Amazon EMR crée et utilise différents groupes de sécurité par défaut pour les clusters d'un sous-réseau privé : pour ElasticMapReduce-Master-Private, ElasticMapReduce-Slave-Private et ElasticMapReduce-

ServiceAccess. Pour plus d'informations, consultez la section Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Pour obtenir une liste complète des listes de contrôle d'accès réseau (listes ACL réseau) de votre cluster, choisissez Security groups for Master (Groupes de sécurité pour principal) et Security groups for Core & Task (Groupes de sécurité pour noyau et tâche) sur la page Cluster Details (Détails de cluster) de la console Amazon EMR.

L'image suivante montre comment un cluster EMR est configuré dans un sous-réseau privé. La seule communication en dehors du sous-réseau est la communication vers Amazon EMR.

L'image suivante représente un exemple de configuration pour un cluster EMR au sein d'un sous-réseau privé connecté à une instance NAT située dans un sous-réseau public.

Sous-réseaux partagés

Le partage de VPC permet aux clients de partager des sous-réseaux avec d'autres comptes AWS au sein de la même organisation AWS. Vous pouvez lancer des clusters Amazon EMR dans des sous-réseaux publics et privés partagés, avec les restrictions suivantes.

Le propriétaire du sous-réseau doit partager un sous-réseau avec vous pour que vous puissiez lancer un cluster Amazon EMR dans celui-ci. Cependant, des sous-réseaux partagés peuvent devenir non partagés ultérieurement. Pour plus d'informations, consultez Utilisation de VPC partagés. Lorsqu'un cluster est lancé dans un sous-réseau partagé qui devient ensuite non partagé, vous pouvez observer des comportements spécifiques en fonction de l'état du cluster Amazon EMR lorsque le sous-réseau devient non partagé.

- Le sous-réseau devient non partagé avant que le cluster soit lancé - Si le propriétaire cesse de partager l'Amazon VPC ou le sous-réseau alors que le participant lance un cluster, il se peut que le cluster ne puisse pas démarrer ou soit partiellement initialisé sans mettre en service toutes les instances demandées.
- Le sous-réseau devient non partagéaprès que le cluster est lancé - Lorsque le propriétaire cesse de partager un sous-réseau ou un Amazon VPC avec le participant, les clusters du participant ne peuvent pas être redimensionnés pour ajouter de nouvelles instances ou remplacer des instances défectueuses.

Lorsque vous lancez un cluster Amazon EMR, plusieurs groupes de sécurité sont créés. Dans un sous-réseau partagé, le participant au sous-réseau contrôle ces groupes de sécurité. Le propriétaire du sous-

réseau peut voir ces groupes de sécurité, mais ne peut pas exécuter d'actions sur ceux-ci. Si le propriétaire du sous-réseau souhaite supprimer ou modifier le groupe de sécurité, le participant qui a créé le groupe de sécurité doit effectuer l'action.

Contrôle des autorisations VPC avec IAM

Par défaut, tous les utilisateurs IAM peuvent consulter l'ensemble des sous-réseaux du compte, et n'importe quel utilisateur peut lancer un cluster dans n'importe quel sous-réseau.

Lorsque vous lancez un cluster dans un VPC, vous pouvez utiliser AWS Identity and Access Management (IAM) pour contrôler l'accès aux clusters et limiter les actions à l'aide de stratégies, de la même manière qu'avec les clusters lancés dans EC2-Classic. Pour plus d'informations sur IAM, consultez IAM Guide de l'utilisateur.

Vous pouvez également utiliser IAM pour contrôler les personnes autorisées à créer et gérer des sous-réseaux. Par exemple, vous pouvez créer un compte d'utilisateur pour administrer les sous-réseaux et un deuxième compte d'utilisateur qui peut lancer des clusters mais ne peut pas modifier Amazon VPC des paramètres. Pour plus d'informations sur la gestion des stratégies et des actions dans Amazon EC2 et Amazon VPC, consultez Stratégies IAM pour Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Configuration d'un VPC pour héberger des clusters

Avant de pouvoir lancer des clusters dans un VPC, vous devez créer un VPC et un sous-réseau. Pour les sous-réseaux publics, vous devez créer une passerelle Internet et l'attacher au sous-réseau. Les instructions suivantes expliquent comment créer un VPC capable d'héberger des clusters Amazon EMR.

Pour créer un sous-réseau pour exécuter des clusters Amazon EMR

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans la barre de navigation, sélectionnez la région dans laquelle exécuter votre cluster.
3. Choisissez Start VPC Wizard.
4. Choisissez la configuration de VPC en sélectionnant l'une des options suivantes :
   - VPC avec un seul sous-réseau public—Sélectionnez cette option si les données utilisées dans le cluster sont disponibles sur Internet (par exemple, dans Amazon S3 ou Amazon RDS).
   - VPC avec des sous-réseaux publics et privés, et un accès VPN matériel—Sélectionnez cette option pour utiliser un sous-réseau privé ou si les données de votre application sont stockées dans votre propre réseau (par exemple, dans une base de données Oracle). Cette option vous permet également d'inclure des sous-réseaux publics au sein du même VPC en tant que sous-réseaux privés.
5. Confirmez les paramètres de VPC. Les images montrent le scénario avec un seul sous-réseau public et le scénario avec des sous-réseaux publics et privés.

- Pour utiliser Amazon EMR, le VPC avec un sous-réseau public doit avoir une passerelle Internet et un sous-réseau.

  Pour un VPC dans un sous-réseau privé, toutes les instances EC2 doivent avoir au moins un chemin vers Amazon EMR via l'interface réseau Elastic. Dans la console, ce paramètre est configuré automatiquement.
- Utilisez un espace d'adresse IP privée pour votre VPC afin de garantir la résolution correcte du nom d'hôte DNS. Dans le cas contraire, vous pouvez être confronté à des défaillances du cluster Amazon EMR. Les plages d'adresses IP suivantes sont incluses :
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
  - 192.168.0.0 - 192.168.255.255
- Choisissez Utiliser plutôt une instance NAT et sélectionnez les options souhaitées.
- Vous pouvez également choisir Ajout de points de terminaison pour S3 à vos sous-réseaux.

- Vérifiez que l'option Activer les noms d'hôte DNS est cochée. Vous avez la possibilité d'activer les noms d'hôte DNS lorsque vous créez un VPC. Pour modifier le paramètre des noms d'hôte DNS, sélectionnez votre VPC dans la liste des VPC, puis sélectionnez Modifier dans le volet des détails. Pour créer une entrée DNS qui n'inclut pas de nom de domaine, créez une valeur pour Jeu d'options DHCP, puis associez-la à votre VPC. Vous ne pouvez pas modifier le nom de domaine à l'aide de la console une fois que le jeu d'option DNS a été créé.

  Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.
- Une bonne pratique concernant Hadoop et les applications connexes consiste à garantir la résolution du nom de domaine complet (FQDN) pour les nœuds. Pour garantir une résolution DNS correcte, configurez un VPC qui inclut un jeu d'options DHCP dont les paramètres sont définis sur les valeurs suivantes :
  - nom-domaine = **`ec2.internal`**

    Utilisez **`ec2.internal`** si votre région est USA Est (Virginie du Nord). Pour les autres régions, utilisez *`region-name`***`.compute.internal`**. Pour des exemples dans `us-west-2`, utiliser **`us-west-2.compute.internal`**. Pour le AWS GovCloud (US-West) région, utiliser **`us-gov-west-1.compute.internal`**.
  - domain-name-servers (serveurs-nom-domaine) = **`AmazonProvidedDNS`**

  Pour plus d'informations, consultez Jeux d'options DHCP dans le Amazon VPC Guide de l'utilisateur.

6. Sélectionnez Créer VPC. Si vous créez une instance NAT, la création peut prendre quelques minutes.

Une fois que le VPC est créé, accédez à la page Sous-réseaux et notez l'identifiant de l'un des sous-réseaux de votre VPC. Vous utilisez ces informations lorsque vous lancez le cluster EMR dans le VPC.

Lancement de clusters dans un VPC

Une fois que vous disposez d'un sous-réseau configuré pour héberger des clusters Amazon EMR, lancez le cluster dans ce sous-réseau en spécifiant l'identifiant du sous-réseau associé lors de la création du cluster.

> **Note**
>
> Amazon EMR prend en charge les sous-réseaux privés à partir de la version 4.2.

Lorsque le cluster est lancé, Amazon EMR ajoute des groupes de sécurité différents si le cluster est lancé dans un sous-réseau VPC public ou privé. Tous les groupes de sécurité autorisent l'infiltration au port 8443 pour communiquer au service Amazon EMR, mais les plages d'adresses IP varient pour les sous-réseaux publics et privés. Amazon EMR gère tous ces groupes de sécurité et peut avoir besoin d'ajouter plus d'adresses IP à la plage AWS au fil du temps. Pour plus d'informations, consultez la section Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Pour gérer le cluster sur un VPC, Amazon EMR lie un périphérique réseau au nœud maître et le gère par le biais de ce dispositif. Vous pouvez afficher cet appareil à l'aide de l' Amazon EC2 Action d'API `DescribeInstances`. Si vous modifiez ce périphérique de quelque manière que ce soit, le cluster peut échouer.

Pour lancer un cluster dans un VPC à l'aide de la console Amazon EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Dans la section Configuration du matériel, pour Réseau, sélectionnez l'ID d'un réseau VPC que vous avez créé précédemment.

5. Pour Sous-réseau EC2, sélectionnez l'ID d'un sous-réseau que vous avez créé précédemment.

   a. Si votre sous-réseau privé est correctement configuré avec les options relatives à l'instance NAT et au point de terminaison S3, il affiche la mention (EMR Ready) (EMR prêt) au-dessus des noms et des identifiants du sous-réseau.

   b. Si votre sous-réseau privé n'a pas d'instance NAT et/ou de point de terminaison S3, vous pouvez y remédier en choisissant Add S3 endpoint and NAT instance (Ajouter point de terminaison S3 et instance NAT), Add S3 endpoint (Ajouter point de terminaison S3) ou Add NAT instance (Ajouter instance NAT). Sélectionnez les options souhaitées pour votre instance NAT et votre point de terminaison S3 et choisissez Configurer.

      Important

      Pour créer une instance NAT à partir d'Amazon EMR, vous avez besoin des autorisations ec2:CreateRoute, `ec2:RevokeSecurityGroupEgress`, `ec2:AuthorizeSecurityGroupEgress`, `cloudformation:DescribeStackEvents` et `cloudformation:CreateStack`.

      Note

      Des frais supplémentaires s'appliquent au lancement d'une instance EC2 pour votre périphérique NAT.

6. Procédez à la création du cluster.

Pour lancer un cluster dans un VPC à l'aide de l'AWS CLI

Note

L'AWS CLI ne permet pas de créer une instance NAT automatiquement et de la connecter à votre sous-réseau privé. Cependant, pour créer un point de terminaison S3 dans votre sous-réseau, vous pouvez utiliser les commandes de l'interface de ligne de commande Amazon VPC. Utilisez la console pour créer des instances NAT et lancer des clusters dans un sous-réseau privé.

Après avoir configuré votre VPC, vous pouvez y lancer des clusters EMR en utilisant la sous-commande `create-cluster` avec le paramètre `--ec2-attributes`. Utilisez le paramètre `--ec2-attributes` pour spécifier le sous-réseau VPC pour votre cluster.

- Pour créer un cluster dans un sous-réseau spécifique, tapez la commande suivante, remplacez *myKey* avec le nom de votre paire de clés EC2 et remplacez *77XXXX03* avec votre ID de sous-réseau.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --
applications Name=Hadoop Name=Hive Name=Pig --use-default-roles --ec2-attributes
 KeyName=myKey,SubnetId=subnet-77XXXX03 --instance-type m5.xlarge --instance-count 3
```

  Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud maître est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utilisent le type d'instance spécifié dans la commande.

  Note

  Si vous n'avez pas encore créé le rôle de service Amazon EMR par défaut et le profil d'instance EC2, tapez `aws emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez l'AWS CLI.

Stratégie Amazon S3 minimum pour un sous-réseau privé

Pour les sous-réseaux privés, vous devez au minimum permettre à Amazon EMR d'accéder aux référentiels Amazon Linux. Avec Amazon EMR 5.25.0 ou version ultérieure, pour activer l'accès en un clic au serveur d'historique Spark permanent, vous devez autoriser Amazon EMR à accéder au compartiment système qui collecte les journaux d'événements Spark. Pour plus d'informations, consultez Accès en un clic au serveur d'historique Spark permanent.

Il vous appartient de déterminer les restrictions de stratégie répondant à vos besoins métier. Par exemple, vous pouvez spécifier la région packages.us-east-1.amazonaws.com pour éviter tout nom de compartiment S3 ambigu. L'exemple de stratégie suivant fournit des autorisations pour accéder aux référentiels Amazon Linux et au compartiment système EMR pour la collecte des journaux d'événements Spark. Remplacez *MyRegion* avec la région où résident vos compartiments de journaux, par exemple `us-east-1`.

```
{
   "Version": "2008-10-17",
   "Statement": [
       {
           "Sid": "AmazonLinuxAMIRepositoryAccess",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": [
               "arn:aws:s3:::packages.*.amazonaws.com/*",
               "arn:aws:s3:::repo.*.amazonaws.com/*"
           ]
       },
       {
           "Sid": "EnableApplicationHistory",
           "Effect": "Allow",
           "Principal": "*",
           "Action": [
               "s3:Put*",
               "s3:Get*",
               "s3:Create*",
               "s3:Abort*",
               "s3:List*"
           ],
           "Resource": [
               "arn:aws:s3:::prod.MyRegion.appinfo.src/*"
           ]
       }
   ]
}
```

Ressources supplémentaires pour en savoir plus sur les VPC

Utilisez les rubriques suivantes pour en savoir plus sur les VPC et les sous-réseaux.

- Sous-réseaux privés dans un VPC
  - Scénario 2: VPC avec sous-réseaux publics et privés (NAT)
  - Instances NAT
  - Haute disponibilité pour les instances NAT Amazon VPC : Un exemple
- Sous-réseaux publics dans un VPC
  - Scénario 1: VPC avec un seul sous-réseau public
- Informations générales sur les VPC
  - Amazon VPC Guide de l'utilisateur
  - Appairage de VPC

- Utilisation d'Elastic Network Interfaces avec votre VPC
- Se connecter en sécurité aux instances Linux s'exécutant dans un VPC privé

Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes

Lorsque vous créez un cluster et spécifiez la configuration du nœud maître, des nœuds principaux et des nœuds de tâches, vous avez deux options de configuration. Vous pouvez utiliser des parcs d'instances ou des groupes d'instances uniformes. L'option de configuration que vous choisissez s'applique à tous les nœuds pour la durée de vie du cluster, et les parcs d'instances ainsi que les groupes d'instances ne peuvent pas coexister dans un cluster. La configuration des parcs d'instances est disponible dans les versions 4.8.0 et ultérieures d'Amazon EMR, à l'exception des versions 5.0.x.

Pour créer une la console EMR, l'AWS CLI ou l'API EMR pour créer des clusters avec l'une ou l'autre de ces configurations. Lorsque vous utilisez la commande `create-cluster` depuis l'AWS CLI, vous utilisez les paramètres `--instance-fleets` pour créer le cluster à l'aide de parcs d'instances ou bien, vous utilisez les paramètres `--instance-groups` pour le créer à l'aide de groupes d'instances uniformes.

Ceci est vrai si vous utilisez l'API EMR. Vous utilisez la configuration `InstanceGroups` pour indiquer une grappe d'objets `InstanceGroupConfig`, ou vous utiliser la configuration `InstanceFleets` pour spécifier une grappe d'objets `InstanceFleetConfig`.

Sur la console EMR, si vous utilisez les paramètres Quick Options (Options rapides) par défaut lors de la création d'un cluster, Amazon EMR applique la configuration des groupes d'instances uniformes au cluster et utilise des instances à la demande. Pour utiliser des instances Spot avec des groupes d'instances uniformes ou pour configurer des parcs d'instances et d'autres personnalisations, choisissez Options avancées.

> Tip
>
> Pour répliquer rapidement et facilement un cluster que vous avez déjà créé, Amazon EMR vous offre deux options sur la console. Vous pouvez cloner le cluster ou générer une commande CLI `create cluster`. Tout d'abord, choisissez Liste de clusters, puis sélectionnez le cluster que vous voulez répliquer. Choisissez Exporter AWS CLI pour qu'Amazon EMR génère la commande CLI `create cluster` équivalente pour le cluster, que vous pouvez ensuite copier et coller. Choisissez le bouton Clone (Cloner) pour qu'Amazon EMR réplique votre configuration de console. Amazon EMR vous présente la dernière étape des Options avancées pour confirmer la configuration du cluster. Vous pouvez choisir Créer un cluster pour créer le nouveau cluster (avec le même nom et un ID de cluster différent) ou vous pouvez choisir Précédent pour revenir et modifier les paramètres.

Parcs d'instances

La configuration des parcs d'instances offre la plus grande variété d'options de mise en service pour les instances EC2. Chaque type de nœud a un parc à instance unique et le parce d'instances de tâches est facultatif. Pour chaque parc d'instances, vous spécifiez jusqu'à cinq types d'instances (si vous utilisez une stratégie d'allocation, jusqu'à 15 types d'instances sur les parcs d'instances de tâches), qui peuvent être provisionnés en tant qu'instances à la demande et instances Spot. Pour les parcs d'instances principaux et de tâches, vous attribuez un capacité cible pour les instances à la demande et une autre pour les instances Spot. Amazon EMR choisit n'importe quel mélange des types d'instance spécifiés pour remplir les capacités cibles, en provisionnant à la fois les instances à la demande et les instances Spot.

Pour le type de nœud maître, Amazon EMR choisit un type d'instance unique dans votre liste d'instances, et vous spécifiez s'il est provisionné en tant qu'instance à la demande ou instance Spot. Les parcs d'instances fournissent également des options supplémentaires pour les achats d'instances Spot et à la demande. Les options d'instances Spot incluent une durée définie (également appelée bloc d'instances

Spot), un délai d'expiration qui spécifie une action à effectuer si la capacité Spot ne peut pas être provisionnée et une stratégie d'allocation préférée (optimisée pour la capacité) pour le lancement des parcs d'instances Spot. Les parcs d'instances à la demande peuvent également être lancés à l'aide d'une stratégie d'allocation (prix le plus bas). Pour plus d'informations, consultez la section Configuration de parcs d'instances (p. 149). Si vous utilisez un rôle de service qui n'est pas le rôle de service par défaut EMR ou la stratégie gérée par EMR dans votre rôle de service, vous devez ajouter des autorisations supplémentaires au rôle de service de cluster personnalisé pour activer la stratégie d'allocation. Pour plus d'informations, consultez la section Service Role for Amazon EMR (EMR Role) (p.).

Groupes d'instances uniformes

Les groupes d'instances uniformes offrent une configuration simplifiée. Chaque cluster Amazon EMR peut inclure jusqu'à 50 groupes d'instances : un groupe d'instances maître qui contient une instance EC2, un groupe d'instances principal qui contient une ou plusieurs instances EC2 et jusqu'à 48 groupes d'instances de tâches facultatifs. Chaque groupe d'instances principal et de tâches peut contenir un nombre quelconque d'instances EC2. Vous pouvez dimensionner chaque groupe d'instances en ajoutant et en retirant des instances EC2 manuellement, ou vous pouvez définir un dimensionnement automatique. Pour plus d'informations sur la configuration des groupes d'instances uniformes, consultez Configuration de groupes d'instances uniformes (p. 159). Pour plus d'informations sur l'ajout et le retrait d'instances, consultez Dimensionnement des ressources de cluster (p. 384).

Rubriques

- Configuration de parcs d'instances (p. 149)
- Configuration de groupes d'instances uniformes (p. 159)

Configuration de parcs d'instances

La configuration des parcs d'instances pour un cluster offre la plus grande variété d'options de mise en service pour les instances EC2. Avec les parcs d'instances, vous spécifiez des capacités cibles pour les instances à la demande et les instances Spot de chaque parc. Lorsque le cluster se lance, Amazon EMR met en service des instances jusqu'à ce que les cibles soient atteintes. Vous pouvez spécifier jusqu'à cinq types d'instance EC2 par parc devant être utilisés par Amazon EMR pour atteindre les cibles. Vous pouvez également sélectionner plusieurs sous-réseaux pour les différentes zones de disponibilité. Quand Amazon EMR lance le cluster, il recherche dans ces sous-réseaux les instances et les options d'achat que vous spécifiez.

Lorsqu'un cluster est en cours d'exécution, si Amazon EC2 récupère une instance Spot en raison d'une augmentation de prix, ou si une instance échoue, Amazon EMR essaie de remplacer l'instance par l'un des types d'instance que vous spécifiez. Il est ainsi plus facile de récupérer de la capacité de lors d'un pic de tarification Spot. Les parcs d'instances vous permettent de développer une stratégie de ressources flexible et élastique pour chaque type de nœud. Par exemple, au sein de parcs spécifiques, vous pouvez avoir un cœur de capacité à la demande complété par une capacité Spot moins chère, le cas échéant, puis passer à la capacité à la demande si la capacité Spot n'est pas disponible à votre prix.

> **Note**
>
> La configuration des parcs d'instances est disponible uniquement dans les versions 4.8.0 et ultérieures d'Amazon EMR, à l'exception des versions 5.0.0 et 5.0.3.

Les stratégies facultatives d'allocation d'instances à la demande et Spot sont disponibles dans Amazon EMR version 5.12.1 et ultérieure.

En tant qu'amélioration de la configuration par défaut des parcs d'instances EMR, la fonction de stratégie d'allocation est disponible dans EMR version 5.12.1 et versions ultérieures. Il optimise l'allocation de la capacité du parc d'instances et vous permet de choisir une stratégie cible pour chaque nœud de cluster.

- Les instances à la demande utilisent une stratégie au prix le plus bas, qui lance d'abord les instances au prix le plus bas.

- Les instances Spot utilisent une stratégie optimisée pour la capacité, qui lance des instances Spot à partir de groupes d'instances Spot ayant une capacité optimale pour le nombre d'instances qui lancent.

L'option de stratégie d'allocation vous permet également de spécifier jusqu'à quinze types d'instance EC2 par nœud de tâche lors de la création de votre cluster, par opposition à cinq maximum autorisés par la configuration du parc d'instances de cluster EMR par défaut.

Récapitulatif des fonctions clés

- Un parc d'instances, et un seul, par type de nœud (maître, principal, tâche). Jusqu'à cinq types d'instance EC2 spécifiés pour chaque parc (15 types par parc d'instances de tâches, si vous utilisez l'option de stratégie d'allocation).
- Amazon EMR choisit une ou l'ensemble des cinq types d'instance EC2 à mettre en service pour les options d'achat Spot et à la demande.
- Établissez des capacités cibles pour les instances Spot et à la demande pour le parc principal et le parc de tâches. Utilisez vCPU ou une unité générique attribuée à chaque instance EC2 qui compte vers les cibles. Amazon EMR provisionne les instances jusqu'à ce que chaque capacité de cible soit complètement comblée. Pour le parc maître, la cible est toujours un seul.
- Choisissez un sous-réseau (zone de disponibilité) ou une plage. Amazon EMR alloue de la capacité dans la zone de disponibilité qui convient le mieux.
- Lorsque vous spécifiez une capacité cible pour instances Spot :
  - Spécifiez un prix Spot maximum pour chaque type d'instance. Amazon EMR provisionne les instances Spot si le prix Spot est plus faible que le maximum spécifié pour Spot. Vous payez le prix Spot, et non le prix Spot maximum.
  - Vous pouvez également spécifier une durée définie (également appelée bloc d'instances Spot) pour chaque parc. Les instances Spot sont suspendues uniquement à l'expiration de la durée définie.
  - Pour chaque parc, définissez une période d'expiration pour la mise en service des instances Spot. Si Amazon EMR ne peut pas mettre en service la capacité Spot, vous pouvez supprimer le cluster ou passer à l'allocation de capacité à la demande à la place.
- Pour chaque flotte, vous pouvez éventuellement choisir d'appliquer des stratégies d'allocation -tarif le plus bas pour les instances à la demande ; optimisation de capacité pour les instances Spot.

Options de parc d'instances

Suivez les instructions suivantes pour comprendre les options de parc d'instances.

Définition des capacités cibles

Spécifiez les capacités cibles que vous souhaitez pour le parc principal et le parc de tâches. Cela permet de déterminer le nombre d'instances à la demande et d'instances Spot mises en service par Amazon EMR. Lorsque vous spécifiez une instance, vous choisissez dans quelle mesure chaque instance compte dans la cible. Lorsqu'une instance à la demande est mise en service, elle est prise en compte dans la cible à la demande. Il en va de même pour les instances Spot. Contrairement aux parcs principaux et de tâches, le parc maître est toujours une seule instance. Ainsi, la capacité cible pour ce parc est toujours un seul.

Lorsque vous utilisez la console, les vCPU du type d'instance EC2 sont pris en compte pour les capacités cibles par défaut. Vous pouvez les remplacer par des Unités génériques, puis spécifier un nombre pour chaque type d'instance EC2. Lorsque vous utilisez la AWS CLI, vous attribuez manuellement des unités génériques à chaque type d'instance.

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide d'AWS Management Console, le nombre de processeurs virtuels affiché pour chaque type d'instance correspond au nombre de cœurs virtuels YARN pour ce type d'instance, et non au nombre de processeurs virtuels EC2 pour ce type

d'instance. Pour de plus amples informations sur le nombre de processeurs virtuels pour chaque type d'instance, veuillez consulter Types d'instances Amazon EC2.

Pour chaque parc, vous spécifiez jusqu'à cinq types d'instance EC2. Si vous utilisez l'option de stratégie d'application, vous pouvez spécifier jusqu'à quinze types d'instance EC2 pour un parc d'instances de tâches. Amazon EMR choisit n'importe quelle combinaison de ces types d'instance EC2 pour remplir vos capacités cibles. Amazon EMR cherche à atteindre complètement la capacité cible. Il est donc possible qu'un excédent se produise. Par exemple, si deux unités non satisfaites sont présentes, et si Amazon EMR peut uniquement mettre en service une instance avec un compte de cinq unités, l'instance est toujours mise en service, et la capacité cible est donc dépassée de trois unités.

Si vous réduisez la capacité cible pour redimensionner un cluster en cours d'exécution, Amazon EMR tente de compléter les tâches d'application et supprime des instances afin d'atteindre la nouvelle cible. Pour plus d'informations, consultez Mise hors service lors de l'achèvement de la tâche (p. 416). Amazon EMR dispose d'un délai de 60 minutes pour effectuer une opération de redimensionnement. Dans certains cas, des tâches peuvent rester en cours d'exécution dans un nœud après 60 minutes, et Amazon EMR indique que l'opération de redimensionnement a réussi et que la nouvelle cible n'a pas été atteinte.

Options de lancement

Pour Instances Spot, vous pouvez spécifier un Prix Spot maximum pour chacun des cinq types d'instance d'une flotte. Vous pouvez régler ce prix soit en pourcentage du prix à la demande, soit en montant en dollars. Amazon EMR provisionne les instances Spot si le prix Spot actuel dans une zone de disponibilité est plus faible que votre prix Spot maximum. Vous payez le prix Spot, et non le prix Spot maximum.

Vous pouvez spécifier une Durée définie pour les instances Spot d'un parc. Lorsque le prix Spot change, Amazon EMR attend que la Durée définie arrive à expiration avant de supprimer les instances. La tarification pour une durée définie s'applique lorsque vous sélectionnez cette option. Si vous ne spécifiez pas de durée définie, les instances sont résiliées dès que le prix Spot dépasse le prix Spot maximum. Pour plus d'informations, consultez Définition d'une durée pour les instances Spot et Tarification des instances Spot Amazon EC2 pour connaître la tarification pour une durée définie.

Pour chaque parc, vous définissez également un Délai dépassé pour la mise en service. Le délai s'applique lorsque le cluster met en service la capacité lors de sa création et qu'il ne peut pas mettre en service suffisamment d'instances Spot pour atteindre la capacité cible en fonction de vos spécifications. Vous spécifiez la période d'expiration et l'action à effectuer. Vous pouvez suspendre le cluster ou passer à l'allocation de capacité à la demande pour atteindre la capacité d'instance Spot restante. Lorsque vous choisissez de passer aux instances à la demande, la capacité Spot restante est ajoutée à la capacité à la demande cible après l'expiration du délai.

Disponible dans Amazon EMR 5.12.1 et versions ultérieures, vous avez la possibilité de lancer des parcs d'instances Spot et à la demande avec une allocation de capacité optimisée. Cette option de stratégie d'allocation peut être définie dans la console de gestion AWS ou à l'aide de l'API `RunJobFlow`. La stratégie d'allocation nécessite des autorisations de rôle de service supplémentaires. Si vous utilisez le rôle de service EMR par défaut et la stratégie gérée (EMR_DefaultRole et AmazonElasticMapReduceRole) pour le cluster, les autorisations pour la stratégie d'allocation sont déjà incluses. Si vous n'utilisez pas le rôle de service EMR par défaut et la stratégie gérée , vous devez les ajouter pour utiliser cette option. Veuillez consulter Service Role for Amazon EMR (EMR Role) (p.).

Pour plus d'informations sur les instances Spot, consultez Instances Spot dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Pour plus d'informations sur les instances à la demande, consultez Instances à la demande dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Options de sous-réseau multiples (zones de disponibilité)

Lorsque vous utilisez plusieurs parcs d'instances, vous pouvez spécifier plusieurs sous-réseaux EC2 au sein d'un VPC, chacun correspondant à une zone de disponibilité différente. Si vous utilisez EC2-Classic, vous spécifiez explicitement la Zone de disponibilité. Amazon EMR identifie la meilleure Zone de disponibilité pour lancer des instances en fonction des spécifications de votre parc. Les instances

sont toujours mises en service dans une seule zone de disponibilité. Vous pouvez sélectionner des sous-réseaux privés ou publics, mais vous ne pouvez pas combiner les deux et les sous-réseaux que vous spécifiez doivent se trouver au sein du même VPC.

Configuration de nœud principal

Étant donné que la flotte d'instances maître est uniquement une instance unique, sa configuration est légèrement différente des parcs d'instances principaux et de tâches. Vous sélectionnez uniquement un parc d'instances maîtres à la demande ou Spot car elle se compose d'une seule instance. Si vous utilisez la console pour créer la flotte d'instances, la capacité cible pour l'option d'achat que vous sélectionnez est définie sur 1. Si vous utilisez l'AWS CLI, définissez toujours `TargetSpotCapacity` ou `TargetOnDemandCapacity` sur 1, le cas échéant. Vous pouvez toujours choisir jusqu'à cinq types d'instance pour le parc d'instances maître. Toutefois, à la différence des parcs d'instance principaux et de tâches, où Amazon EMR peut mettre en service plusieurs instances de différents types, Amazon EMR sélectionne un seul type d'instance à mettre en service pour le parc d'instances principal.

Utilisation de la console pour configurer des parcs d'instances

Pour créer un cluster à l'aide de parcs d'instances, utilisez la configuration Options avancées sur la console Amazon EMR.

Avec EMR version 5.12.1 et versions ultérieures, la méthode préférée pour créer un parc d'instances de cluster consiste à appliquer des stratégies d'allocation. Cette nouvelle option est recommandée pour une mise en service plus rapide du cluster, une allocation d'instances Spot plus précise et moins d'interruptions d'instances Spot par rapport à un parc d'instances sans la nouvelle option de stratégie d'allocation. La création d'un cluster à l'aide de la nouvelle option de stratégie d'allocation nécessite plusieurs autorisations qui incluent automatiquement le rôle de service EMR par défaut et la stratégie gérée par EMR (EMR_DefaultRole et AmazonElasticMapReduceRole). Si vous utilisez un rôle de service personnalisé ou une stratégie gérée pour votre cluster, vous devez ajouter les nouvelles autorisations suivantes pour la stratégie d'allocation avant de créer le cluster. Veuillez consulter Service Role for Amazon EMR (EMR Role) (p. 218).

```
"ec2:DeleteLaunchTemplate", "ec2:CreateLaunchTemplate",
"ec2:DescribeLaunchTemplates", "ec2:CreateFleet"
```

Pour créer un cluster avec des parcs d'instance à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Ouvrez la console Amazon EMR à l'adresse suivante : ,.
4. Choisissez Créer un cluster.
5. En haut de la fenêtre de la console, choisissez Aller aux options avancées, saisissez Configuration du logiciel options, puis choisissez Suivant.
6. En dessous Composition du cluster, choisissez Parcs d'instances.
7. Pour Réseau, saisissez une valeur. Si vous choisissez un VPC à côté de Réseau, choisissez un seul Sous-réseau EC2 ou utilisez CTRL + clic pour choisir plusieurs sous-réseaux EC2. Les sous-réseaux que vous sélectionnez doivent être du même type (public ou privé). Si vous en choisissez un seul, votre cluster se lance dans ce sous-réseau. Si vous choisissez un groupe, c'est le sous-réseau du groupe le mieux adapté qui est sélectionné lors du lancement du cluster.

   > **Note**
   >
   > Votre compte et la région peuvent vous offrir la possibilité de choisir l'option Lancer dans EC2-Classic à côté de Réseau. Si vous choisissez cette option, choisissez une ou plusieurs Zones de disponibilité EC2 au lieu de Sous-réseaux EC2. Pour de plus amples informations, consultez Amazon EC2 et Amazon VPC dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

8. En dessous Stratégie d'allocation, cochez la case pour appliquer les stratégies d'allocation.

   Si vous ne souhaitez pas utiliser la nouvelle option de stratégie d'allocation, laissez la case décochée.
9. Pour chaque Type de nœud, si vous souhaitez modifier le nom par défaut d'un parc d'instances, cliquez sur l'icône en forme de crayon, puis saisissez un nom convivial. Si vous souhaitez supprimer le Tâche parc d'instances, cliquez sur l'icône X à droite de la ligne Tâche.
10. Cliquez sur Ajouter/supprimer des types d'instance au parc et choisissez jusqu'à cinq types d'instance dans la liste pour les parcs d'instances principaux et principaux ; ajoutez jusqu'à quinze types d'instance pour les parcs d'instances de tâches. Amazon EMR peut choisir de provisionner n'importe quel mélange de ces types d'instance lorsqu'il lance le cluster.
11. Pour chaque type d'instance de noyau et de tâche, choisissez la façon dont vous souhaitez définir la capacité pondérée (Chaque instance compte comme X unités) pour cette instance. Le nombre de vCores YARN pour chaque type d'instance de parc est utilisé comme unités de capacité pondérées par défaut, mais vous pouvez modifier la valeur pour toutes les unités qui ont du sens pour vos applications.
12. En dessous Capacité cible, définissez le nombre total d'instances à la demande et Spot que vous souhaitez par parc. EMR s'assurera que les instances de la flotte remplissent les unités demandées pour la capacité cible ponctuelle et à la demande. Si aucune unité à la demande ou Spot n'est spécifiée pour une flotte, aucune capacité n'est allouée pour cette flotte.
13. Si une flotte est configurée avec une capacité cible pour Spot, vous pouvez entrer votre prix Spot maximum en tant que % de la tarification à la demande, ou vous pouvez entrer un montant en dollars ($) en USD.
14. Pour que les volumes EBS soient attachés au type d'instance lors de sa mise en service, cliquez sur le crayon à côté de Stockage sur EBS et indiquez les options de configuration EBS.
15. Si vous avez établi un compte instantané pour Point, choisissez Options Spot avancées conformément aux directives suivantes :

    - Durée définie—si laissé à Non défini (par défaut), les instances Spot se terminent dès que le prix Spot dépasse le prix Spot maximum, ou lorsque le cluster résilie. Si vous définissez une valeur, les instances Spot ne sont pas suspendues tant que la durée n'est pas arrivée à expiration.

      Important

      Si vous définissez une Durée définie, une tarification pour une durée définie spéciale s'applique. Pour plus de détails sur la tarification, consultez Tarification des instances Spot Amazon EC2.
    - Délai dépassé pour la mise en service—Ces paramètres permettent de contrôler ce que fait Amazon EMR lorsqu'il ne peut pas mettre en service des instances Spot au sein des Types d'instances du parc que vous spécifiez. Vous saisissez un délai d'expiration en minutes, puis choisissez de Mettre fin au cluster ou de Passer à une mise en service des instances à la demande. Si vous choisissez de basculer vers des instances à la demande, la capacité attribuée des instances à la demande compte pour la capacité cible des instances Spot, et Amazon EMR alloue les instances à la demande jusqu'à ce que la capacité cible pour les instances Spot soit satisfaite.
16. Choisir Suivant, modifiez les autres paramètres du cluster, puis cliquez sur Suivant.
17. Si vous avez choisi d'appliquer la nouvelle option de stratégie d'allocation, dans le champ Options de sécurité paramètres, sélectionnez un Rôle EMR et Profil d'instance EC2 qui contiennent les autorisations requises pour l'option de stratégie d'allocation. Sinon, la création du cluster échouera.
18. Cliquez sur Créer un cluster.

Utilisation de l'interface de ligne de commande pour configurer des parcs d'instances

- Pour créer et lancer un cluster avec des parcs d'instances, utilisez la commande `create-cluster` avec les paramètres `--instance-fleet`.

- Pour afficher les détails de configuration des parcs d'instances dans un cluster, utilisez la commande `list-instance-fleets`.
- Pour apporter des modifications à la capacité cible d'un parc d'instances, utilisez la commande `modify-instance-fleet`.
- Pour ajouter un parc d'instances de tâches à un cluster qui n'en possède pas déjà, utilisez la commande `add-instance-fleet`.
- Pour utiliser l'option de stratégie d'allocation lors de la création d'un parc d'instances, mettez à jour le rôle de service pour inclure l'exemple de document de stratégie ci-dessous.

Example Document de stratégie pour le rôle de service

Il s'agit des autorisations de rôle de service supplémentaires requises pour créer un cluster qui utilise l'option de stratégie d'allocation de parc d'instances.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteLaunchTemplate",
    "ec2:CreateLaunchTemplate",
    "ec2:DescribeLaunchTemplates",
    "ec2:CreateFleet"
   ],
   "Resource": "*"
  }
```

> **Note**
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

Création d'un cluster avec la configuration de parcs d'instances

Les exemples suivants illustrent les commandes `create-cluster` avec une variété d'options que vous pouvez combiner.

> **Note**
>
> Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, utilisez `aws emr create-default-roles` pour les créer avant d'utiliser la commande `create-cluster`.

Example Exemple: Maître à la demande, cœur à la demande avec type d'instance unique, VPC par défaut

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
 \
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
```

Example Exemple: Maître d'instance Spot, noyau d'instance Spot avec type d'instance unique, VPC par défaut

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets
 InstanceFleetType=MASTER,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0
 \
InstanceFleetType=CORE,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}
```

Example Exemple: Maître à la demande, cœur mixte avec type d'instance unique, sous-réseau EC2 unique

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c'] \
--instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
 \
InstanceFleetType=CORE,TargetOnDemandCapacity=2,TargetSpotCapacity=6,InstanceTypeConfigs=['{InstanceTyp
```

Example Exemple: Maître à la demande, cœur d'instance Spot avec plusieurs types d'instances pondérées, durée définie et délai d'expiration pour l'instance Spot, plage de sous-réseaux EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c','subnet-
de67890f'] \
--instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
 \
InstanceFleetType=CORE,TargetSpotCapacity=11,InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5
\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{BlockDurationMinutes=180,TimeoutDurationMinutes=120,TimeoutAc
```

Example Exemple: Maître à la demande, noyau mixte et tâche avec plusieurs types d'instances pondérées, durée définie et délai d'expiration pour les instances Spot principales, plage de sous-réseaux EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c','subnet-
de67890f'] \
--instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
 \
InstanceFleetType=CORE,TargetOnDemandCapacity=8,TargetSpotCapacity=6,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{BlockDurationMinutes=180,TimeoutDurationMinutes=120,TimeoutAc
 \
InstanceFleetType=TASK,TargetOnDemandCapacity=3,TargetSpotCapacity=3,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}']
```

Example Exemple: Maître Spot, pas de noyau ou de tâche, configuration EBS, VPC par défaut

```
aws emr create-cluster --release-label emr 5.3.1 -service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets InstanceFleetType=MASTER,TargetSpotCapacity=1,\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=60,TimeoutAction=TERMINATE_CLUSTER}'},
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,\
EbsConfiguration={EbsOptimized=true,EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=gp2,
\
SizeIn GB=100}},{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iop
 s=100},VolumesPerInstance=4}]}}']
```

Example Utilisation d'un fichier de configuration JSON

Vous pouvez configurer des paramètres de parc d'instances dans un fichier JSON, puis référencer le fichier JSON en tant que seul paramètre pour les parcs d'instances. Par exemple, la commande suivante référence un fichier de configuration JSON, my-fleet-config.json :

```
aws emr create-cluster --release-label emr-5.30.0 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets file://my-fleet-config.json
```

my-fleet-config.json spécifie les parcs d'instances maîtres, principaux et de tâches, comme illustré dans l'exemple suivant. Le parc d'instances principal utilise un prix Spot maximum (`BidPrice`) sous la forme d'un pourcentage du prix à la demande, alors que les parcs d'instances de tâches et maîtres utilisent un prix Spot maximum (BidPriceAsPercentageofOnDemandPrice) sous la forme d'une somme en USD.

```
[
    {
        "Name": "Masterfleet",
        "InstanceFleetType": "MASTER",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "SWITCH_TO_ON_DEMAND"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    },
    {
        "Name": "Corefleet",
        "InstanceFleetType": "CORE",
        "TargetSpotCapacity": 1,
        "TargetOnDemandCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price"
        },
            "SpotSpecification": {
                "AllocationStrategy": "capacity-optimized",
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
```

```
            {
                "InstanceType": "m5.xlarge",
                "BidPriceAsPercentageOfOnDemandPrice": 100
            }
        ]
    },
    {
        "Name": "Taskfleet",
        "InstanceFleetType": "TASK",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    }
]
```

Obtention des détails de configuration des parcs d'instances dans un cluster

Utilisez la commande `list-instance-fleets` pour obtenir les détails de configuration des parcs d'instances dans un cluster. La commande utilise un ID de cluster en entrée. 'exemple suivant illustre une la commande et sa sortie pour un cluster qui contient un groupe d'instances de tâches maître et un groupe d'instances de tâches principal. Pour la syntaxe complète, consultez ListInstanceFleets dans le Amazon EMR API Reference.

```
list-instance-fleets --cluster-id 'j-12ABCDEFGHI34JK'
```

```
{
    "InstanceFleets": [
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759094.637,
                    "CreationDateTime": 1488758719.817
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 6,
            "Name": "CORE",
            "InstanceFleetType": "CORE",
            "LaunchSpecifications": {
                "SpotSpecification": {
                    "TimeoutDurationMinutes": 60,
                    "TimeoutAction": "TERMINATE_CLUSTER"
                }
            },
            "ProvisionedOnDemandCapacity": 2,
            "InstanceTypeSpecifications": [
```

```
                {
                    "BidPrice": "0.5",
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 2
                }
            ],
            "Id": "if-1ABC2DEFGHIJ3"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759058.598,
                    "CreationDateTime": 1488758719.811
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 0,
            "Name": "MASTER",
            "InstanceFleetType": "MASTER",
            "ProvisionedOnDemandCapacity": 1,
            "InstanceTypeSpecifications": [
                {
                    "BidPriceAsPercentageOfOnDemandPrice": 100.0,
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 1
                }
            ],
           "Id": "if-2ABC4DEFGHIJ4"
        }
    ]
}
```

Modification des capacités cibles pour un parc d'instances

Utilisez la commande `modify-instance-fleet` pour spécifier les nouvelles capacités cibles d'un parc d'instances. Vous devez spécifier l'ID de cluster et l'ID de parc d'instances. Utilisez la commande `list-instance-fleets` pour extraire les ID de parc d'instances.

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' /
--instance-fleet
 InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

Ajout d'un parc d'instances de tâches à un cluster

Si un cluster n'a que des parcs d'instances maître et principaux, vous pouvez utiliser la commande `add-instance-fleet` pour ajouter un parce d'instances de tâches. Vous pouvez l'utiliser pour ajouter des parcs d'instances de tâches.

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetType=TASK,TargetSpotCapacity=1,/
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=20,TimeoutAction=TERMINATE_CLUSTER}'},
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Configuration de groupes d'instances uniformes

Avec la configuration des groupes d'instances, chaque type de nœud (maître, principal ou tâche) se compose du même type d'instance et de la même option d'achat pour les instances : À la demande ou Spot. Vous spécifiez ces paramètres lorsque vous créez un groupe d'instances. Ils ne peuvent pas être modifiés ultérieurement. Vous pouvez, cependant, ajouter des instances du même type et une option d'achat aux groupes d'instances principaux et de tâches. Vous pouvez aussi supprimer des instances.

Pour ajouter des types d'instances différents après la création d'un cluster, vous pouvez ajouter des groupes d'instances de tâches supplémentaires. Vous pouvez choisir différents types d'instances et différentes options d'achat pour chaque groupe d'instances. Pour plus d'informations, consultez la section Dimensionnement des ressources de cluster (p. 384).

Cette section couvre la création d'un cluster avec des groupes d'instances uniformes. Pour plus d'informations sur la modification d'un groupe d'instances existant par l'ajout ou le retrait d'instances manuellement ou avec un dimensionnement automatique, consultez Manage Clusters (p. 319).

Utilisation de la console pour configurer des groupes d'instances uniformes

La procédure suivante couvre les Options avancées lors de la création d'un cluster. L'utilisation des Quick options (Options rapides) permet aussi de créer un cluster avec la configuration des groupes d'instance. Pour plus d'informations sur l'utilisation des Quick Options (Options rapides), consultez le Didacticiel de mise en route.

Pour créer un cluster avec des groupes d'instances uniformes à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées, saisissez les options Configuration des logiciels, puis choisissez Suivant.
4. Dans l'écran Configuration du matériel, conservez l'option Groupes d'instances uniformes sélectionnée.
5. Choisissez le Réseau, puis le Sous-réseau EC2 pour votre cluster. Le sous-réseau que vous choisissez est associé à un groupe de disponibilité, répertorié dans chaque sous-réseau. Pour plus d'informations, consultez la section Configuration de la mise en réseau (p. 137).

   > **Note**
   >
   > Votre compte et la région peuvent vous offrir la possibilité de choisir l'option Lancer dans EC2-Classic à côté de Réseau. Si vous choisissez cette option, choisissez une Zone de disponibilité EC2 au lieu d'un Sous-réseau EC2. Pour de plus amples informations, consultez Amazon EC2 et Amazon VPC dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

6. Au sein de chaque ligne Type de nœud :
   - Sous Type de nœud, si vous voulez modifier le nom par défaut du groupe d'instances, cliquez sur l'icône en forme de crayon et saisissez un nom convivial. Si vous voulez retirer le groupe d'instances Tâche, cliquez sur l'icône X. Choisissez Add task instance group (Ajouter groupe d'instances de tâche) pour ajouter des groupes d'instances supplémentaires de Task (Tâche).
   - Sous Instance type (Type d'instance), cliquez sur l'icône en forme de crayon et choisissez le type d'instance que vous voulez utiliser pour ce type de nœud.

     > **Important**
     >
     > Lorsque vous choisissez un type d'instance à l'aide d'AWS Management Console, le nombre de processeurs virtuels affiché pour chaque type d'instance correspond au nombre

de cœurs virtuels YARN pour ce type d'instance, et non au nombre de processeurs virtuels EC2 pour ce type d'instance. Pour de plus amples informations sur le nombre de processeurs virtuels pour chaque type d'instance, veuillez consulter Types d'instances Amazon EC2.

- Sous Instance type (Type d'instance), cliquez sur l'icône en forme de crayon en regard de Configurations, puis modifiez les configurations des applications pour chaque groupe d'instances.
- Sous Nombre d'instances, saisissez le nombre d'instances à utiliser pour chaque type de nœud.
- Sous Option d'achat, choisissez À la demande ou Spot. Si vous choisissez Spot, sélectionnez une option pour le prix maximum des instances Spot. Par défaut, l'option Utiliser le prix à la demande comme prix maximal est sélectionnée. Vous pouvez sélectionner Définir le prix maximal $/h, puis saisir votre prix maximum. La zone de disponibilité du Sous-réseau EC2 que vous choisissez se trouve en dessous du Prix spot maximum.

  Tip

  Survolez l'info-bulle d'informations à côté de Spot pour voir le prix Spot actuel de toutes les zones de disponibilités dans la région en cours. Le prix Spot le plus bas figure en vert. Vous pouvez utiliser ces informations pour changer de Sous-réseau EC2.
- Sous Auto Scaling for Core and Task node types (Auto Scaling pour les types de nœuds de tâche et de noyau), choisissez l'icône en forme de crayon, puis configurez les options de scalabilité automatique (auto scaling). Pour plus d'informations, consultez la section Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401).

7. Choisissez Ajouter un groupe d'instances de tâches et configurez les paramètres comme décrit à l'étape précédente.
8. Choisissez Suivant, modifiez les autres paramètres du cluster, puis lancez le cluster.

Utilisation de l'AWS CLI pour créer un cluster avec des groupes d'instances uniformes

Pour spécifier la configuration des groupes d'instances pour un cluster à l'aide de l'AWS CLI, utilisez la commande `create-cluster` avec le paramètre `--instance-groups`. Amazon EMR présume l'option d'achat à la demande, sauf si vous spécifiez l'argument `BidPrice` pour un groupe d'instances. Pour obtenir des exemples de commandes `create-cluster` qui permettent de lancer des groupes d'instances uniformes avec des instances à la demande et une variété d'options de cluster, saisissez `aws emr create-cluster help` dans la ligne de commande, ou consultez la section create-cluster dans le AWS CLI Command Reference.

Vous pouvez utiliser l'AWS CLI pour créer des groupes d'instances uniformes dans un cluster qui utilise des instances Spot. Le prix Spot proposé varie en fonction de la zone de disponibilité. Lorsque vous utilisez l'interface de ligne de commande ou l'API, vous pouvez spécifier la zone de disponibilité à l'aide de l'argument `AvailabilityZone` (si vous utilisez un réseau EC2-classic) ou de l'argument `SubnetID` du paramètre `--ec2-attributes` . La zone de disponibilité ou le sous-réseau que vous sélectionnez s'appliquent au cluster ; ils sont donc utilisés pour tous les groupes d'instances. Si vous ne spécifiez pas une zone de disponibilité ou un sous-réseau de manière explicite, Amazon EMR sélectionne la zone de disponibilité au prix Spot le plus faible lors du lancement du cluster.

L'exemple suivant illustre une commande `create-cluster` qui crée des groupes d'instances maîtres, principaux et deux groupes d'instances de tâches qui utilisent tous des instances Spot. Remplacez *`myKey`* avec le nom de votre paire de clés EC2.

Note

Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "MySpotCluster" --release-label emr-5.30.1 \
--use-default-roles --ec2-attributes KeyName=myKey \
--instance-groups
 InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,BidPrice=0.25 \
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.03 \
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=4,BidPrice=0.03 \
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.04
```

Utilisation du kit SDK Java pour créer un groupe d'instances

Vous instanciez un objet `InstanceGroupConfig` qui spécifie la configuration d'un groupe d'instances pour un cluster. Pour utiliser des instances Spot, vous définissez les propriétés `withBidPrice` et `withMarket` sur l'objet `InstanceGroupConfig`. Le code suivant montre comment définir des groupes d'instances maîtres, principales et de tâches qui exécutent des instances Spot.

```
InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
 .withInstanceCount(1)
 .withInstanceRole("MASTER")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.25");

InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
 .withInstanceCount(4)
 .withInstanceRole("CORE")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.03");

InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
 .withInstanceCount(2)
 .withInstanceRole("TASK")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.10");
```

Consignes et bonnes pratiques pour la configuration de cluster

Utilisez les consignes de cette section pour déterminer les types d'instance, les options d'achat, ainsi que la quantité de stockage à allouer à chaque type de nœud dans un cluster EMR.

Quel type d'instance dois-je utiliser ?

Vous pouvez ajouter des instances EC2 à votre cluster de différentes façons, selon que vous utilisez ou non la configuration des groupes d'instances ou la configuration des parcs d'instances pour le cluster.

- Groupes d'instances
  - Ajoutez manuellement des instances du même type aux groupes d'instances principaux et de tâches existants.
  - Ajoutez manuellement un groupe d'instances de tâches pouvant utiliser un type d'instance différent.
  - Définissez une scalabilité automatique dans Amazon EMR pour un groupe d'instances, qui ajoute ou supprime automatiquement des instances en fonction de la valeur d'une métrique Amazon CloudWatch que vous spécifiez. Pour plus d'informations, consultez la section Dimensionnement des ressources de cluster (p. 384).
- Parcs d'instances

- Ajoutez un parc d'instances de tâches unique.
- Modifiez la capacité cible des instances à la demande et des instances Spot pour les parcs d'instances principaux et de tâches existants. Pour plus d'informations, consultez la section Configuration de parcs d'instances (p. 149).

Une manière de planifier les instances de votre cluster consiste à exécuter un cluster de test avec un ensemble représentatif d'échantillons de données et à surveiller l'utilisation des nœuds dans le cluster. Pour plus d'informations, consultez la section View and Monitor a Cluster (p. 319). Une autre méthode consiste à calculer la capacité des instances que vous envisagez et à comparer cette valeur à la taille de vos données.

En général, le type de nœud maître, qui attribue les tâches, n'a pas besoin d'une instance EC2 dotée d'une grande puissance de traitement. Les instances EC2 du type de nœud principal, qui traitent les tâches et stockent les données dans HDFS, ont besoin d'une puissance de traitement et d'une capacité de stockage. Les instances EC2 du type de nœud de tâches, qui ne stockent pas de données, n'ont besoin que d'une puissance de traitement. Pour connaître les consignes relatives aux instances EC2 disponibles et leur configuration, consultez Configurer les instances EC2 (p. 131).

Les consignes suivantes s'appliquent à la plupart des clusters Amazon EMR.

- Le nœud maître n'a pas de grandes exigences en matière de calcul. Pour la plupart des clusters de 50 nœuds ou moins, envisagez d'utiliser une instance m5.xlarge. Pour les clusters de plus de 50 nœuds, envisagez d'utiliser un m4.xlarge, car les vCores dans m4.xlarge sont deux fois plus grands que m5.xlarge.
- Les besoins en calcul des nœuds principaux et de tâches dépendent du type de traitement que votre application effectue. De nombreuses tâches peuvent être exécutées sur les types d'instance m5.xlarge, qui offrent des performances équilibrées en termes d'UC, d'espace disque et d'E/S. Si votre application a des dépendances externes qui entraînent des retards (par exemple, l'indexation de site web pour collecter des données), vous pouvez éventuellement exécuter le cluster sur des instances t2.medium pour réduire les coûts alors que les instances attendent la fin des dépendances. Pour améliorer les performances, envisagez d'exécuter le cluster à l'aide d'instances m4.xlarge pour les nœuds principaux et de tâches. Si des phases différentes de votre cluster ont des exigences de capacité différentes, vous pouvez commencer avec un petit nombre de nœuds principaux et augmenter ou diminuer le nombre de nœuds de tâches pour satisfaire aux exigences de capacité variables de votre flux de travail.
- La plupart des clusters Amazon EMR peuvent s'exécuter sur les types d'instance EC2 Standard tels que m5.xlarge et m4.xlarge. Les clusters nécessitant des calculs intensifs peuvent bénéficier d'une exécution sur des instances à CPU intensif, qui ont proportionnellement plus de ressources CPU que de RAM. Les applications de base de données et de mise en cache en mémoire peuvent bénéficier d'une exécution sur des instances à mémoire élevée. Les applications nécessitant beaucoup de ressources CPU et réseau, telles que les applications d'analyse, NLP et de Machine Learning, peuvent bénéficier d'une exécution sur des instances Cluster Compute, qui fournissent proportionnellement des ressources d'UC intensives et des performances réseau accrues.
- La quantité de données que vous pouvez traiter dépend de la capacité de vos nœuds principaux et de la taille de vos données en entrée, au cours du traitement et en sortie. Les ensembles de données d'entrée, intermédiaires et de sortie résident tous sur le cluster au cours du traitement.
- Par défaut, le nombre total d'instances EC2 que vous pouvez exécuter sur un seul compte AWS est 20. Cela signifie que le nombre total de nœuds que vous pouvez avoir dans un cluster est de 20. Pour plus d'informations sur la façon de demander une augmentation de cette limite pour votre compte, consultez Limites AWS.

Quand faut-il utiliser des instances Spot ?

Lorsque vous lancez un cluster dans Amazon EMR, vous pouvez choisir de lancer des instances principales, de noyau ou de tâche sur des instances Spot. Comme chaque type de groupe d'instances

joue un rôle différent dans le cluster, il existe des implications du lancement de chaque type de nœud sur des instances Spot. Vous ne pouvez pas modifier l'option d'achat d'une instance lorsque le cluster est en cours d'exécution. Pour modifier un groupe d'instances à la demande en instances Spot ou inversement, vous devez suspendre le cluster et en lancer un nouveau pour les nœuds principaux et de noyau. Pour les nœuds de tâches, vous pouvez lancer un nouveau groupe d'instances de tâche ou un parc d'instances et supprimer l'ancien.

Rubriques

- Paramètres Amazon EMR pour empêcher l'échec de tâche en raison d'une résiliation d'instance Spot de nœud de tâche (p. 163)
- Nœud principal sur une instance Spot (p. 164)
- Nœuds de noyau sur les instances Spot (p. 164)
- Nœuds de tâche sur les instances Spot (p. 164)
- Configurations d'instances pour les scénarios d'application (p. 165)

Paramètres Amazon EMR pour empêcher l'échec de tâche en raison d'une résiliation d'instance Spot de nœud de tâche

Les instances Spot étant souvent utilisées pour exécuter des nœuds de tâche, Amazon EMR dispose d'une fonctionnalité par défaut pour la planification de tâches YARN de sorte que les tâches en cours d'exécution n'échouent pas lorsque des nœuds de tâche en cours d'exécution sur des instances Spot sont résiliées. Amazon EMR s'en assure en n'autorisant les processus maîtres des applications à s'exécuter uniquement sur des nœuds principaux. Le processus maître de l'application contrôle l'exécution des tâches et doit rester actif pendant toute la durée de vie de la tâche.

Amazon EMR version 5.19.0 et ultérieures utilise la fonction des étiquettes de nœud YARN intégrés pour y parvenir. (Les versions antérieures utilisaient un correctif de code). Les propriétés des classifications de configuration `yarn-site` et `capacity-scheduler` sont configurées par défaut de sorte que le planificateur de YARN et le planificateur d'équité tirent parti des étiquettes de nœud. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de sorte que les masters de l'application ne soient planifiés que sur des nœud avec l'étiquette CORE. La modification manuelle de propriétés apparentées dans les classifications de configuration yarn-site et capacity-scheduler ou directement dans les fichiers XML associés pourrait rompre cette fonctionnalité ou la modifier.

Amazon EMR configure les variables et propriétés suivantes par défaut. Faites attention lorsque vous configurez ces propriétés.

- yarn-site (yarn-site.xml) sur tous les nœuds
  - `yarn.node-labels.enabled: true`
  - `yarn.node-labels.am.default-node-label-expression: 'CORE'`
  - `yarn.node-labels.fs-store.root-dir: '/apps/yarn/nodelabels'`
  - `yarn.node-labels.configuration-type: 'distributed'`
- yarn-site (yarn-site.xml) sur les nœuds principaux et de noyau
  - `yarn.nodemanager.node-labels.provider: 'config'`
  - `yarn.nodemanager.node-labels.provider.configured-node-partition: 'CORE'`
- capacity-scheduler (capacity-scheduler.xml) sur tous les nœuds
  - `yarn.scheduler.capacity.root.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.accessible-node-labels.CORE.capacity: 100`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels.CORE.capacity: 100`

Note

À partir de la série Amazon EMR version 6.x, la fonction des étiquettes de nœud YARN est désactivée par défaut. Les processus principaux des applications peuvent s'exécuter à la fois sur les nœuds de noyau et sur les nœuds de tâche par défaut. Vous pouvez activer la fonction d'étiquetage des nœuds YARN en configurant les propriétés suivantes :

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Nœud principal sur une instance Spot

Le nœud maître contrôle et dirige le cluster. Lorsque le cluster est arrêté, il prend fin, si bien que vous devez lancer uniquement le nœud principal en tant qu'instance Spot si vous exécutez un cluster où un arrêt soudain est acceptable. Ce peut être le cas, si vous testez une nouvelle application, si vous avez un cluster qui conserve périodiquement des données dans un magasin externe tel que Amazon S3, ou si vous exécutez un cluster où le coût est plus important que l'exécution du cluster jusqu'à la fin.

Lorsque vous lancez le groupe d'instances maître en tant qu'instance Spot, le cluster ne démarre pas tant que cette demande d'instance Spot n'est pas satisfaite. Ceci doit être pris en compte lorsque vous sélectionnez votre prix Spot maximum.

Vous pouvez uniquement ajouter un nœud maître d'instances Spot lorsque vous lancez le cluster. Il n'est pas possible d'ajouter ni de supprimer de nœuds maîtres à partir d'un cluster en cours d'exécution.

En général, il suffirait d'exécuter le nœud maître en tant qu'instance Spot si vous exécutez l'ensemble du cluster (tous les groupes d'instances) sous forme d'instances Spot.

Nœuds de noyau sur les instances Spot

Les nœuds principaux traitent les données et stockent les informations à l'aide de HDFS. La résiliation d'une instance de noyau risque la perte de données. Pour cette raison, vous devez uniquement exécuter des nœuds de noyau sur des instances Spot ou des pertes de données HDFS partielles sont tolérables.

Lorsque vous lancez le groupe d'instances principal sous forme d'instances Spot, Amazon EMR attend de pouvoir mettre en service toutes les instances principales demandées avant de lancer le groupe d'instances. En d'autres termes, si vous demandez six instances Amazon EC2 et seulement cinq d'entre elles sont disponibles au même prix Spot maximum (ou moins), le groupe d'instances ne s'exécutera pas. Amazon EMR continue d'attendre jusqu'à ce que les six instances Amazon EC2 soient disponibles ou jusqu'à ce que vous résiliez le cluster. Vous pouvez modifier le nombre d'instances Spot dans un groupe d'instances de noyau pour ajouter de la capacité à un cluster en cours d'exécution. Pour plus d'informations sur la façon de travailler avec des groupes d'instances et comment les instances Spot fonctionnent avec des parcs d'instances, consultez : the section called “Configurer des parcs ou groupes d'instances” (p. 148).

Nœuds de tâche sur les instances Spot

Les nœuds de tâches traitent les données, mais ne conservent pas de données persistantes dans HDFS. S'ils sont suspendus car le prix Spot a augmenté et dépasse votre prix Spot maximum, aucune donnée n'est perdue et l'effet sur votre cluster est minimal.

Lorsque vous lancez un ou plusieurs groupes d'instances de tâches sous forme d'instances Spot, Amazon EMR met en service autant de nœuds de tâches que possible à votre prix Spot maximum. Cela signifie que si vous demandez un groupe d'instances de tâches à six nœuds et que seules cinq instances Spot sont disponibles à votre prix Spot maximum ou sous ce prix, Amazon EMR lance le groupe d'instances avec cinq nœuds et ajoute le sixième plus tard, si cela est possible.

Le lancement des groupes d'instances de tâches sous forme d'instances Spot est un moyen stratégique d'étendre la capacité de votre cluster tout en réduisant au maximum les coûts. Si vous lancez vos groupes d'instances maîtres et principaux en tant qu'instances à la demande, leur capacité est garantie pour l'exécution du cluster. Vous pouvez ajouter des instances de tâches à vos groupes d'instances de tâches en fonction des besoins, afin de gérer les pics de trafic ou d'accélérer le traitement de données.

Vous pouvez ajouter ou supprimer des nœuds de tâches à l'aide de la console, de l'AWS CLI ou de l'API. Vous pouvez également ajouter des groupes de tâches supplémentaires, mais vous ne pouvez pas supprimer un groupe de tâches après sa création.

Configurations d'instances pour les scénarios d'application

Le tableau suivant est une référence rapide pour les options et configurations de types de nœuds qui sont généralement adaptés à plusieurs scénarios d'applications. Cliquez sur le lien pour afficher des informations supplémentaires sur chaque type de scénario.

Scénario d'application	Option d'achat de nœud principal	Option d'achat de nœud de noyau	Option d'achat de nœud de tâche
Entrepôts de données et clusters de longue durée (p. 165)	À la demande	Combinaison d'instances à la demande ou de parc d'instances	Combinaison d'instances Spot ou de parc d'instances
Charges de travail axées sur les coûts (p. 165)	Spot	Spot	Spot
Charges de travail essentielles pour les données (p. 165)	À la demande	À la demande	Combinaison d'instances Spot ou de parc d'instances
Tests d'application (p. 166)	Spot	Spot	Spot

Il existe plusieurs scénarios dans lesquels les instances Spot sont utiles pour exécuter un cluster Amazon EMR.

Entrepôts de données et clusters de longue durée

Si vous exécutez un cluster Amazon EMR persistant qui présente une variation prévisible de sa capacité de calcul (tel qu'un entrepôt de données), vous pouvez gérer un pic de demande à moindre coût grâce aux instances Spot. Vous pouvez lancer vos groupes d'instances maîtres et principaux sous forme d'instances à la demande pour gérer la capacité normale et lancer le groupe d'instances de tâches sous forme d'instances Spot pour gérer les exigences des pics de charge.

Charges de travail axées sur les coûts

Si vous exécutez des clusters transitoires pour lesquels une réduction des coûts est plus importante que la durée d'exécution, et qu'une perte partielle de travail est acceptable, vous pouvez exécuter l'ensemble du cluster (groupes d'instances maîtres, principaux et de tâches) sous forme d'instances Spot, afin de bénéficier des plus grandes économies de coûts possibles.

Charges de travail essentielles pour les données

Si vous exécutez un cluster pour lequel une réduction des coûts est plus importante que la durée d'exécution, mais qu'une perte partielle de travail n'est pas acceptable, lancez les groupes d'instances maîtres et principales à la demande et complétez-les avec un ou plusieurs groupes d'instances de tâches

sous forme d'instances Spot. L'exécution des groupes d'instances maîtres et principaux à la demande garantit la conservation de vos données dans HDFS et la protection du cluster contre sont arrêt en raison de fluctuations Spot du marché, tout en offrant des économies de coûts provenant de l'exécution des groupes d'instances de tâches sous forme d'instances Spot.

Tests d'application

Lorsque vous testez une nouvelle application afin de préparer son lancement dans un environnement de production, vous pouvez exécuter l'ensemble du cluster (groupes d'instances maîtres, principaux et de tâches) sous forme d'instances Spot pour réduire les coûts des tests.

Calcul de la capacité HDFS requise pour un cluster

La quantité de stockage HDFS disponible pour votre cluster dépend des facteurs suivants :

- Nombre d'instances EC2 utilisées pour les nœuds principaux.
- Capacité du stockage d'instances EC2 pour le type d'instance utilisé. Pour plus d'informations sur les volumes de stockage d'instance, consultez Stockage d'instance Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
- Nombre et taille des volumes EBS attachés aux nœud principaux.
- Facteur de réplication, qui représente le mode de stockage de chaque bloc de données dans HDFS pour assurer une redondance similaire à RAID. Par défaut, le facteur de réplication est égal à trois pour un cluster composé de 10 nœuds principaux ou plus, à deux pour un cluster de 4 à 9 nœuds principaux, et à un pour un cluster de trois nœuds ou moins.

Pour calculer la capacité HDFS d'un cluster, pour chaque nœud principal, ajoutez la capacité du volume de stockage d'instance à la capacité de stockage EBS (le cas échéant). Multipliez le résultat par le nombre de nœuds principaux, puis divisez le total par le facteur de réplication en fonction du nombre de nœuds principaux. Par exemple, un cluster contenant 10 nœuds principaux de type i2.xlarge qui possèdent 800 Go de stockage d'instance sans volumes EBS associés dispose au total d'environ 2 666 Go disponibles pour HDFS (10 nœuds x 800 Go ÷ facteur de réplication de 3).

Si la valeur de capacité HDFS calculée est inférieure à vos données, vous pouvez augmenter la quantité de stockage HDFS de différentes manières :

- En créant un cluster avec des volumes EBS supplémentaires, ou en ajoutant des groupes d'instances avec des volumes EBS attachés à un cluster existant
- En ajoutant d'autres nœuds principaux
- En choisissant un type d'instance EC2 avec une capacité de stockage supérieure
- En utilisant la compression des données
- En modifiant les paramètres de configuration de Hadoop pour réduire le facteur de réplication

La réduction du facteur de réplication doit être utilisée avec précautions, car elle réduit la redondance des données HDFS et l'aptitude du cluster à récupérer après la perte ou l'endommagement de blocs HDFS.

Configure Cluster Logging and Debugging

Lorsque vous planifiez votre cluster, vous devez déterminer la quantité de support de débogage que vous souhaitez rendre disponible. Lorsque vous commencez à développer votre application de traitement des données, nous vous recommandons de tester l'application sur un cluster traitant un petit sous-ensemble représentatif de vos données. Lorsque vous procédez ainsi, il est probable que vous souhaitiez tirer parti

de tous les outils de débogage que propose Amazon EMR, comme l'archivage des fichiers journaux dans Amazon S3.

Lorsque vous avez terminé la phase de développement et que votre application de traitement des données passe en production, vous pouvez décider de réduire le débogage. Vous pouvez ainsi économiser le coût de stockage des archives de fichiers journaux dans Amazon S3 et réduire la charge de traitement sur le cluster, qui n'a plus besoin d'écrire les états dans Amazon S3. En revanche, en cas de problèmes, vous aurez moins d'outils disponibles pour les traiter.

Default Log Files

Par défaut, chaque cluster écrit les fichiers journaux sur le nœud maître. Ils sont écrits dans le répertoire `/mnt/var/log/`. Vous pouvez y accéder à l'aide de SSH pour vous connecter au nœud maître, comme décrit dans Connexion au nœud maître à l'aide de SSH (p. 368). Étant donné que ces journaux existent sur le nœud principal, ils ne sont plus disponibles lorsque le nœud est hors service, soit à cause de la fermeture du cluster ou à cause d'une erreur.

Vous n'avez pas besoin d'activer d'options pour que les fichiers journaux soient écrits sur le nœud maître. Il s'agit en effet du comportement par défaut d'Amazon EMR et de Hadoop.

Un cluster génère plusieurs types de fichiers journaux, y compris :

- Step logs — These logs are generated by the Amazon EMR service and contain information about the cluster and the results of each step. The log files are stored in `/mnt/var/log/hadoop/steps/` directory on the master node. Each step logs its results in a separate numbered subdirectory: `/mnt/var/log/hadoop/steps/s-`*`stepId1`*`/` for the first step, `/mnt/var/log/hadoop/steps/s-`*`stepId2`*`/`, for the second step, and so on. The 13-character step identifiers (e.g. stepId1, stepId2) are unique to a cluster.
- Hadoop and YARN component logs — The logs for components associated with both Apache YARN and MapReduce, for example, are contained in separate folders in `/mnt/var/log`. The log file locations for the Hadoop components under `/mnt/var/log` are as follows: hadoop-hdfs, hadoop-mapreduce, hadoop-httpfs, and hadoop-yarn. The hadoop-state-pusher directory is for the output of the Hadoop state pusher process.
- Bootstrap action logs — If your job uses bootstrap actions, the results of those actions are logged. The log files are stored in /mnt/var/log/bootstrap-actions/ on the master node. Each bootstrap action logs its results in a separate numbered subdirectory: `/mnt/var/log/bootstrap-actions/1/` for the first bootstrap action, `/mnt/var/log/bootstrap-actions/2/`, for the second bootstrap action, and so on.
- Instance state logs — These logs provide information about the CPU, memory state, and garbage collector threads of the node. The log files are stored in `/mnt/var/log/instance-state/` on the master node.

Archive Log Files to Amazon S3

> **Note**
>
> Vous ne pouvez pas actuellement utiliser l'agrégation des journaux vers Amazon S3 avec l'utilitaire `yarn logs`.

Vous pouvez configurer un cluster afin qu'il archive régulièrement les fichiers journaux stockés sur le nœud maître dans Amazon S3. Vous avez ainsi la garantie que les fichiers journaux sont disponibles une fois que le cluster est mis hors service, qu'il s'agisse d'une fermeture normale ou d'une erreur. Amazon EMR archive les fichiers de journaux sur Amazon S3 à des intervalles de 5 minutes.

Pour que les fichiers journaux soient archivés dans Amazon S3, vous devez activer cette fonctionnalité lorsque vous lancez le cluster. Vous pouvez effectuer cette opération à l'aide de la console, de l'interface de ligne de commande ou de l'API. Par défaut, l'archivage des fichiers est activé pour les clusters lancés à

l'aide de la console. Il doit être activé manuellement pour les clusters lancés dans Amazon S3 à l'aide de l'interface de ligne de commande ou de l'API.

To archive log files to Amazon S3 using the console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Dans le Options générales section, dans le Journalisation , acceptez l'option par défaut : Enabled

   Cela détermine si Amazon EMR capture des données de journal détaillées dans Amazon S3. Vous ne pouvez définir cette option que lors de la création du cluster. Pour plus d'informations, consultez View Log Files (p. 333),
5. Dans le champ Dossier S3 tapez (ou accédez à) un chemin d'accès Amazon S3 pour stocker vos journaux. Vous pouvez également autoriser la console à générer automatiquement un chemin d'accès Amazon S3. Si vous tapez le nom d'un dossier qui n'existe pas dans le compartiment, il est automatiquement créé.

   Lorsque cette valeur est définie, Amazon EMR copie les fichiers journaux des instances EC2 du cluster vers Amazon S3. Cela empêche la perte des fichiers journaux lorsque le cluster prend fin et que les instances EC2 hébergeant le cluster sont arrêtées. Ces journaux sont utiles à des fins de dépannage.

   Pour de plus amples informations, veuillez consulter Afficher les fichiers journaux.
6. Dans le champ Chiffrement des journaux, sélectionnez Chiffrer les journaux stockés dans S3 avec une clé gérée par le client AWS KMS. Sélectionnez ensuite une clé AWS KMS dans la liste ou entrez un ARN de clé. Vous pouvez également créer une nouvelle clé KMS.

   Cette option n'est disponible qu'avec les Amazon EMR versions 5.30.0 et ultérieures. Pour utiliser cette option, ajoutez l'autorisation à KMS pour votre profil d'instance EC2 et votre rôle EMR. Pour plus d'informations, consultez To encrypt log files stored in Amazon S3 with an AWS KMS customer managed key (p. 168),
7. Procédez à la création du cluster, comme décrit dans Plan and Configure Clusters (p. 50).

To encrypt log files stored in Amazon S3 with an AWS KMS customer managed key

Avec Amazon EMR version 5.30.0 et ultérieure (sauf Amazon EMR 6.0.0), vous pouvez chiffrer les fichiers journaux stockés dans Amazon S3 avec une clé gérée par le client AWS KMS. Pour activer cette option dans la console, suivez les étapes de la section To archive log files to Amazon S3 using the console (p. 168). Votre profil d'instance Amazon EC2 et votre rôle Amazon EMR doivent répondre aux conditions préalables suivantes :

- The Amazon EC2 instance profile used for your cluster must have permission to use `kms:GenerateDataKey`.
- The Amazon EMR role used for your cluster must have permission to use `kms:DescribeKey`.
- The Amazon EC2 instance profile and Amazon EMR role must be added to the list of key users for the specified AWS KMS customer managed key, as the following steps demonstrate:
  1. Open the AWS Key Management Service (AWS KMS) console at https://console.aws.amazon.com/kms.
  2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
  3. Select the alias of the CMK to modify.
  4. On the key details page under Key Users, choose Add.
  5. In the Add key users dialog box, select your Amazon EC2 instance profile and Amazon EMR role.

6. Choose Add.

Pour de plus amples informations, veuillez consulter Rôles IAM Service utilisés par Amazon EMR, et Utilisation des stratégies de clé dans le guide du développeur AWS Key Management Service.

To archive log files to Amazon S3 using the AWS CLI

Pour archiver les fichiers journaux dans Amazon S3 à l'aide de l'AWS CLI, tapez la commande `create-cluster` et spécifiez le chemin d'accès au journal Amazon S3 à l'aide du paramètre `--log-uri`.

- Pour enregistrer les fichiers vers Amazon S3 tapez la commande suivante et remplacez *myKey* avec le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --log-uri
 s3://mybucket/logs/ --applications Name=Hadoop Name=Hive Name=Pig --use-default-roles
 --ec2-attributes KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

> **Note**
>
> Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

To aggregate logs in Amazon S3 using the AWS CLI

> **Note**
>
> Actuellement, vous ne pouvez pas utiliser l'agrégation des journaux avec l'utilitaire `yarn logs`. Vous pouvez uniquement utiliser l'agrégation prise en charge par cette procédure.

L'agrégation de journaux (Hadoop 2.x) compile les journaux de tous les conteneurs d'une application individuelle en un seul fichier. Pour activer l'agrégation des journaux dans Amazon S3 à l'aide de l'AWS CLI, vous utilisez une action d'amorçage lors du lancement du cluster pour activer l'agrégation des journaux et spécifier le compartiment dans lequel stocker les journaux.

- > **Important**
>
> Ce paramètre n'était pas opérationnel dans les versions EMR 4.x précédentes. Veuillez utiliser les versions ultérieures à 4.3.0 si vous souhaitez configurer cette option.

  Pour activer l'agrégation de journaux, créez le fichier de configuration `myConfig.json`, qui contient les éléments suivants :

```
[
  {
    "Classification": "yarn-site",
    "Properties": {
      "yarn.log-aggregation-enable": "true",
      "yarn.log-aggregation.retain-seconds": "-1",
      "yarn.nodemanager.remote-app-log-dir": "s3:\/\/mybucket\/logs"
    }
```

```
  }
]
```

Tapez la commande suivante et remplacez *myKey* avec le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.5.0 --applications
 Name=Hadoop --use-default-roles --ec2-attributes KeyName=myKey --instance-type
 m5.xlarge --instance-count 3 --configurations file://./myConfig.json
```

Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

> **Note**
>
> Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Enable the Debugging Tool

L'outil de débogage vous permet d'explorer plus facilement les fichiers journaux depuis la console EMR. Pour plus d'informations, consultez View Log Files in the Debugging Tool (p. 336), Lorsque vous activez le débogage dans un cluster, Amazon EMR archive les fichiers journaux dans Amazon S3, puis les indexe. Vous pouvez ensuite utiliser la console pour rechercher les journaux des étapes, travaux, tâches et tentatives de tâches du cluster de manière intuitive.

Pour pouvoir utiliser l'outil de débogage dans la console EMR, vous devez activer le débogage lorsque vous lancez le cluster à l'aide de la console, de l'interface de ligne de commande ou de l'API.

Pour activer l'outil de débogage à l'aide de la console Amazon EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Dans la section Cluster Configuration (Configuration de cluster), dans le champ Logging (Journalisation), choisissez Activé. Vous ne pouvez pas activer le débogage sans activer la journalisation.
5. Dans le champ Log folder S3 location (Emplacement S3 de dossier de journal), tapez un chemin d'accès Amazon S3 pour stocker vos journaux.
6. Dans le champ Debugging (Débogage), choisissez Activé.

   L'option de débogage crée un échange Amazon SQS pour publier les messages de débogage dans le backend de service Amazon EMR. Des frais peuvent s'appliquent à la publication de messages. Pour plus d'informations, consultez https://aws.amazon.com/sqs,
7. Procédez à la création du cluster, comme décrit dans Plan and Configure Clusters (p. 50).

Pour activer l'outil de débogage à l'aide de l'AWS CLI

Pour activer le débogage à l'aide de l'AWS CLI, tapez la sous-commande `create-cluster` avec le paramètre `--enable-debugging`. Vous devez également spécifier le paramètre `--log-uri` lorsque vous activez le débogage.

- Pour activer le débogage en utilisant AWS CLI, tapez la commande suivante et remplacez *myKey* avec le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.1.0 --log-uri
 s3://mybucket/logs/ --enable-debugging --applications Name=Hadoop Name=Hive Name=Pig
 --use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --
instance-count 3
```

  Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

  Note

  Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Example Activation du débogage à l'aide du kit SDK Java

Activation du débogage à l'aide de StepConfig :

```
    StepFactory stepFactory = new StepFactory();
 StepConfig enabledebugging = new StepConfig()
     .withName("Enable debugging")
     .withActionOnFailure("TERMINATE_JOB_FLOW")
     .withHadoopJarStep(stepFactory.newEnableDebuggingStep());
```

Dans cet exemple, `new StepFactory()` utilise `us-east-1` comme région par défaut. Si votre cluster est lancé dans une autre région, vous devez spécifier la région à l'aide de `new StepFactory("region_name.elasticmapreduce")`, par exemple `new StepFactory("ap-northeast-2.elasticmapreduce")`.

Debugging Option Information

A partir de la version 4.1, Amazon EMR prend en charge le débogage dans toutes les régions.

Amazon EMR crée une file d'attente Amazon SQS pour traiter les données de débogage. Des frais peuvent être facturés pour les messages. Cependant, Amazon SQS propose une offre gratuite pour 1 000 000 demandes maximum. Pour plus d'informations, consultez la page de détails Amazon SQS.

Le débogage nécessite l'utilisation de rôles. Votre profil d'instance et votre rôle de service doivent vous permettre d'utiliser toutes les opérations d'API Amazon SQS. Si vos rôles sont liés aux stratégies gérées par Amazon EMR, vous n'avez pas besoin de faire quoi que ce soit pour modifier vos rôles. Si vous disposez de rôles personnalisés, vous devez ajouter des autorisations `sqs:*`. Pour plus d'informations, consultez Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213),

Tag Clusters

Il peut s'avérer utile de classer vos ressources AWS de différentes manières ; par exemple, par objectif, par propriétaire ou par environnement. Vous pouvez effectuer cela dans Amazon EMR en affectant des métadonnées personnalisées pour vos clusters Amazon EMR à l'aide de balises. Une balise est constituée

d'une clé et d'une valeur que vous définissez. Pour Amazon EMR, le cluster est l'élément situé au niveau des ressources que vous pouvez baliser. Par exemple, vous pouvez définir un ensemble de balises pour les clusters de votre compte, afin de suivre le propriétaire de chaque cluster ou d'identifier un cluster de production par rapport à un cluster de test. Nous vous recommandons de créer un ensemble de balises cohérent pour répondre aux exigences de votre organisation.

Lorsque vous ajoutez une balise à un cluster Amazon EMR, la balise est également appliquée à chaque instance Amazon EC2 active associée au cluster. De même, lorsque vous supprimez une balise d'un cluster Amazon EMR, cette balise est supprimée de chaque instance Amazon EC2 active associée.

> **Important**
>
> Utilisez la console ou l'interface de ligne de commande Amazon EMR pour gérer les balises sur les instances Amazon EC2 qui font partie d'un cluster plutôt que la console ou l'interface de ligne de commande Amazon EC2, car les modifications que vous effectuez dans Amazon EC2 ne sont pas synchronisées dans le système de balisage Amazon EMR.

Vous pouvez identifier une instance Amazon EC2 faisant partie d'un cluster Amazon EMR en recherchant les balises système suivantes. Dans cet exemple : *CORE* est la valeur du rôle du groupe d'instances et *j-12345678* est un exemple de valeur d'identifiant de flux de travail (cluster) :

- aws:elasticmapreduce:instance-group-role=*CORE*
- aws:elasticmapreduce:job-flow-id=*j-12345678*

> **Note**
>
> Amazon EMR et Amazon EC2 interprètent vos balises comme une chaîne de caractères sans signification sémantique.

Vous pouvez gérer les balises à l'aide d'AWS Management Console, de l'interface de ligne de commande et de l'API.

Vous pouvez ajouter des balises lors de la création d'un cluster Amazon EMR et vous pouvez ajouter, modifier ou supprimer des balises d'un cluster Amazon EMR en cours d'exécution. Le concept de modification d'une balise s'applique à la console Amazon EMR. En revanche, si vous utilisez l'interface de ligne de commande et l'API, la modification consiste à supprimer l'ancienne balise et en ajouter une nouvelle. Vous pouvez modifier les clés et valeurs de balise, et vous pouvez supprimer des balises d'une ressource à tout moment pendant l'exécution d'un cluster. Cependant, vous ne pouvez pas ajouter, modifier ou supprimer des balises à partir d'un cluster hors service ou d'instances hors service qui étaient précédemment associées à un cluster qui est toujours actif. Vous pouvez également définir la valeur d'une balise sur une chaîne vide, mais vous ne pouvez pas définir la valeur d'une balise sur null.

Si vous utilisez AWS Identity and Access Management (IAM) avec vos instances Amazon EC2 pour les autorisations basées sur les ressources par balise, vos stratégies IAM sont appliquées à des balises qu'Amazon EMR propage aux instances Amazon EC2 d'un cluster. Pour que les balises Amazon EMR puissent être propagées à vos instances Amazon EC2, votre stratégie IAM pour Amazon EC2 doit accorder les autorisations d'appel des API Amazon EC2 CreateTags et DeleteTags. En outre, les balises propagées peuvent avoir une incidence sur les autorisations basées sur les ressources d'Amazon EC2. Les balises propagées vers Amazon EC2 peuvent être lues en tant que conditions dans votre stratégie IAM, tout comme les autres balises Amazon EC2. Gardez votre stratégie IAM à l'esprit lorsque vous ajoutez des balises à vos clusters Amazon EMR, afin d'éviter que les utilisateurs IAM disposent d'autorisations erronées pour un cluster. Pour éviter les problèmes, assurez-vous que vos stratégies IAM n'incluent pas de conditions sur les balises que vous prévoyez également d'utiliser sur vos clusters Amazon EMR. (Pour de plus amples informations, veuillez consulter Contrôle de l'accès aux ressources d'un Amazon EC2).

Tag Restrictions

Les restrictions de base suivantes s'appliquent aux balises.

- Restrictions that apply to Amazon EC2 resources apply to Amazon EMR as well. For more information, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions.
- Do not use the `aws:` prefix in tag names and values because it is reserved for AWS use. In addition, you cannot edit or delete tag names or values with this prefix.
- You cannot change or edit tags on a terminated cluster.
- A tag value can be an empty string, but not null. In addition, a tag key cannot be an empty string.
- Keys and values can contain any alphabetic character in any language, any numeric character, white spaces, invisible separators, and the following symbols: _ . : / = + - @

Pour plus d'informations sur l'étiquetage à l'aide du AWS Management Console, voir Utilisation des étiquettes dans la console dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Pour plus d'informations sur l'étiquetage à l'aide du Amazon EC2API ou ligne de commande, voir Aperçu API et CLI dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Tag Resources for Billing

Vous pouvez utiliser des balises pour organiser votre facture AWS afin qu'elle tienne compte de votre propre structure de coûts. Pour ce faire, inscrivez-vous pour obtenir votre facture de compte AWS avec les valeurs de clé de balise incluses. Vous pouvez organiser vos informations de facturation par valeurs de clé de balise pour voir le coût de vos ressources combinées. Bien qu'Amazon EMR et Amazon EC2 aient des relevés de facturation différents, les balises sur chaque cluster sont également placées sur chaque instance associée afin que vous puissiez utiliser des balises pour lier les coûts Amazon EMR et Amazon EC2 associés.

Par exemple, vous pouvez baliser plusieurs ressources avec un nom d'application spécifique, puis organiser vos informations de facturation pour afficher le coût total de cette application dans plusieurs services. Pour de plus amples informations, veuillez consulter Répartition et balisage des coûts dans AWS Billing and Cost Management Guide de l'utilisateur.

Add Tags to a New Cluster

Vous pouvez ajouter des balises à un cluster lors de sa création.

Pour ajouter des balises lors de la création d'un nouveau cluster à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Sur le Étape 3 : Paramètres généraux du cluster page, dans le Tags , tapez un Clé pour votre étiquette.

   Lorsque vous commencez à saisir la Clé, une nouvelle ligne s'affiche automatiquement en prévision de la balise suivante.
4. Si vous le souhaitez, saisissez une Valeur pour la balise.
5. Répétez les étapes précédentes pour chaque paire clé-valeur de balise à ajouter au cluster. Lorsque le cluster est lancé, les balises que vous saisissez sont automatiquement associées au cluster.

Pour ajouter des balises lors de la création d'un nouveau cluster à l'aide de l'AWS CLI

L'exemple suivant montre comment ajouter une balise à un nouveau cluster à l'aide de l'AWS CLI. Pour ajouter des balises lors de la création d'un cluster, saisissez la sous-commande `create-cluster` avec le paramètre `--tags`.

- Pour ajouter une balise nommée *costCenter* avec la valeur clé *marketing* lorsque vous créez un cluster, tapez la commande suivante et remplacez *myKey* avec le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --applications
 Name=Hadoop Name=Hive Name=Pig --tags "costCenter=marketing" --use-default-roles --
ec2-attributes KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

  Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud principal est lancé et les instances restantes sont lancées en tant que nœuds principaux. Tous les nœuds utiliseront le type d'instance spécifié dans la commande.

  > **Note**
  >
  > Si vous n'avez pas encore créé le rôle de service EMR par défaut et le profil d'instance EC2, tapez aws `emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Adding Tags to an Existing Cluster

Vous pouvez également ajouter des balises à un cluster existant.

Pour ajouter des balises à un cluster existant à l'aide de la console

1. Dans la console Amazon EMR, sélectionnez la page Liste de clusters et cliquez sur un cluster auquel vous souhaitez ajouter des balises.
2. Sur la page Cluster Details (Détails de cluster), dans le champ Balises, cliquez sur Afficher tout/Modifier.
3. Sur la page Afficher tout/Modifier, cliquez sur Ajouter.
4. Cliquez sur le champ vide dans la colonne Clé et saisissez le nom de votre clé.
5. Vous pouvez également cliquer sur le champ vide dans la colonne Valeur et saisir le nom de votre valeur.
6. Lorsque vous commencez à saisir une nouvelle balise, une autre ligne de balise vide s'affiche sous la balise que vous modifiez. Répétez les étapes précédentes sur la nouvelle ligne de balise pour chaque balise à ajouter.

Pour ajouter des balises à un cluster en cours d'exécution à l'aide de l'AWS CLI

L'exemple suivant montre comment ajouter des balises à un cluster en cours d'exécution à l'aide de l'AWS CLI. Tapez la sous-commande `add-tags` avec le paramètre `--tag` pour attribuer des balises à un identifiant de ressource (ID de cluster). L'ID de ressource est l'identifiant de cluster disponible via la console ou la commande `list-clusters`.

> **Note**
>
> Actuellement, la sous-commande `add-tags` n'accepte qu'un ID de ressource.

- Pour ajouter deux balises à un cluster de course (une avec une clé nommée *production* sans valeur et l'autre avec une clé nommée *costCenter* avec une valeur de *marketing*) tapez la commande suivante et remplacez *j-KT4XXXXXXXX1NM* avec votre ID de cluster.

```
aws emr add-tags --resource-id j-KT4XXXXXXXX1NM --tag "costCenter=marketing" --
tag "other=accounting"
```

Note

Lorsque des balises sont ajoutées à l'aide de l'interface de ligne de commande AWS, la commande n'a aucune sortie.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

View Tags on a Cluster

Si vous souhaitez voir toutes les balises associées à un cluster, vous pouvez les afficher dans la console ou à l'interface de ligne de commande.

Pour afficher les balises sur un cluster à l'aide de la console

1. Dans la console Amazon EMR, sélectionnez la page Liste de clusters et cliquez sur un cluster pour afficher les balises.
2. Sur la page Cluster Details (Détails de cluster), certaines balises sont affichées dans le champ Balises. Cliquez sur Afficher tout/Modifier pour afficher toutes les balises disponibles sur le cluster.

Pour afficher les balises sur un cluster à l'aide de l'AWS CLI

Pour afficher les balises sur un cluster à l'aide de l'AWS CLI, tapez la sous-commande `describe-cluster` avec le paramètre `--query`.

- Pour afficher les balises d'un cluster, tapez la commande suivante et remplacez *`j-KT4XXXXXXXX1NM`* avec votre ID de cluster.

```
aws emr describe-cluster --cluster-id j-KT4XXXXXX1NM --query Cluster.Tags
```

La sortie affiche toutes les informations de balises sur le cluster, sous une forme similaire à celle-ci :

```
Value: accounting     Value: marketing
Key: other            Key: costCenter
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Remove Tags from a Cluster

Si vous n'avez plus besoin une balise, vous pouvez la supprimer du cluster.

Pour supprimer les balises d'un cluster à l'aide de la console

1. Dans la console Amazon EMR, sélectionnez la page Liste de clusters et cliquez sur un cluster duquel vous souhaitez supprimer des balises.
2. Sur la page Cluster Details (Détails de cluster), dans le champ Balises, cliquez sur Afficher tout/Modifier.
3. Dans la boîte de dialogue Afficher tout/Modifier, cliquez sur l'icône X située à côté de la balise à supprimer, puis cliquez sur Enregistrer.
4. (Facultatif) Répétez l'étape précédente pour chaque paire clé-valeur de balise à supprimer du cluster.

Pour supprimer les balises d'un cluster à l'aide de l'AWS CLI

Pour supprimer les balises d'un cluster à l'aide de l'AWS CLI, tapez la sous-commande `remove-tags` avec le paramètre `--tag-keys`. Lorsque vous supprimez une balise, seul le nom de la clé est requis.

- Pour supprimer une balise d'un cluster, tapez la commande suivante et remplacez *j-KT4XXXXXXXX1NM* avec votre ID de cluster.

```
aws emr remove-tags --resource-id j-KT4XXXXXX1NM --tag-keys "costCenter"
```

  Note

  Actuellement, vous ne pouvez pas supprimer plusieurs balises à l'aide d'une seule commande.

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Drivers and Third-Party Application Integration

Vous pouvez exécuter plusieurs applications Big Data populaires sur Amazon EMR avec une tarification à l'usage. Cela signifie que vous payez des frais horaires minimes supplémentaires pour l'application tierce lorsque le cluster est en cours d'exécution. Cela vous permet d'utiliser l'application sans avoir à acquérir une licence annuelle. Les sections suivantes décrivent certains des outils que vous pouvez utiliser avec EMR.

Rubriques

- Use Business Intelligence Tools with Amazon EMR (p. 176)

Use Business Intelligence Tools with Amazon EMR

Vous pouvez utiliser des outils populaires d'aide à la décision, comme Microsoft Excel, MicroStrategy, QlikView et Tableau, avec Amazon EMR pour explorer et visualiser vos données. Un grand nombre de ces outils ont besoin d'un pilote ODBC (Open Database Connectivity) ou JDBC (Java Database Connectivity). Vous pouvez télécharger et installer les pilotes les plus récents du lien ci-dessous :

http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/

Les versions antérieures des conducteurs sont ici :

http://awssupportdatasvcs.com/bootstrap-actions/Simba/

Pour plus d'informations sur la façon de connecter un outil d'intelligence d'entreprise comme Microsoft Excel à Hive, consultez http://cdn.simba.com/products/Hive/doc/Simba_Hive_ODBC_Quickstart.pdf.

Sécurité dans Amazon EMR

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture 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-même. Le modèle de responsabilité partagée décrit ceci comme sécurité du cloud et sécurité dans le cloud :

- La 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 programmes de conformité AWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon EMR, consultez 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 la sensibilité de vos données, les exigences de votre entreprise,et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Amazon EMR. Lorsque vous développez des solutions sur Amazon EMR, utilisez les technologies suivantes pour sécuriser les ressources et les données de cluster en fonction des besoins de votre entreprise. Les rubriques de cette section vous montrent comment configurer Amazon EMR et utiliser d'autres services AWS pour répondre à vos objectifs de sécurité et de conformité.

Configurations de la sécurité

Les configurations de la sécurité dans Amazon EMR sont des modèles de différentes configurations de la sécurité. Vous pouvez créer une configuration de sécurité à réutiliser facilement pour une configuration de sécurité à chaque fois que vous créez un cluster. Pour plus d'informations, consultez Utilisation de configurations de sécurité pour configurer la sécurité du cluster (p. 179).

Protection des données

Vous pouvez implémenter le chiffrement des données pour aider à protéger les données au repos dans Amazon S3 et dans le stockage d'instances de cluster, ainsi que les données en transit. Pour de plus amples informations, veuillez consulter Chiffrer les données au repos et en transit (p. 197).

AWS Identity and Access Management avec Amazon EMR

AWS Identity and Access Management (IAM) est un service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux ressources AWS. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (disposant des autorisations) à utiliser les ressources Amazon EMR. IAM est un service AWS que vous pouvez utiliser sans frais supplémentaires.

- Les stratégies basées sur l'identité IAM – IAM autorisent ou refusent les autorisations permettant aux utilisateurs et aux groupes IAM d'effectuer des actions. Ces stratégies peuvent être combinées avec un balisage pour contrôler l'accès cluster par cluster. Pour de plus amples informations, veuillez consulter AWS Identity and Access Management pour Amazon EMR (p. 207).

- Rôles IAM – Le rôle de service Amazon EMR, le profil d'instance et le rôle lié au service contrôlent la façon dont Amazon EMR peut accéder à d'autres services AWS. Pour de plus amples informations, veuillez consulter Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).
- Rôles IAM pour les demandes EMRFS à Amazon S3 – Lorsque Amazon EMR accède à Amazon S3, vous pouvez spécifier le rôle IAM à utiliser en fonction de l'utilisateur, du groupe ou de l'emplacement des données EMRFS dans Amazon S3. Ceci vous permet de contrôler précisément le cluster si les utilisateurs peuvent accéder à des fichiers depuis l'intérieur Amazon EMR. Pour de plus amples informations, veuillez consulter Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234).

Kerberos

Vous pouvez configurer Kerberos pour fournir une authentification renforcée par le biais du chiffrement par clé secrète. Pour plus d'informations, consultez Use Kerberos Authentication (p. 255).

Lake Formation

Vous pouvez utiliser les autorisations Lake Formation avec le AWS Glue Data Catalog pour fournir un accès précis au niveau des colonnes aux bases de données et aux tables du catalogue de données AWS Glue. Lake Formation permet l'authentification unique fédérée aux blocs-notes EMR ou Apache Zeppelin à partir d'un système d'identité d'entreprise. Pour plus d'informations, consultez Intégration Amazon EMR avec AWS Lake Formation (Bêta) (p. 282).

Secure Socket Shell (SSH)

SSH offre aux utilisateurs un moyen sécurisé de se connecter à l'interface de ligne de commande sur les instances de cluster. Il fournit également un tunneling permettant d'afficher des interfaces web hébergées par des applications sur le nœud principal. Les clients peuvent s'authentifier à l'aide de Kerberos ou d'une paire de clés Amazon EC2. Pour de plus amples informations, veuillez consulter Use an Amazon EC2 Key Pair for SSH Credentials (p. 255) et Connexion au cluster (p. 367).

Groupes de sécurité Amazon EC2

Les groupes de sécurité agissent comme pare-feu virtuel pour les instances de cluster EMR, en limitant le trafic réseau entrant et sortant. Pour plus d'informations, consultez Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Mises à jour de l'AMI Amazon Linux par défaut pour Amazon EMR

Lorsqu'une instance Amazon EC2 dans un cluster qui est basé sur l'AMI Amazon Linux par défaut pour Amazon EMR démarre pour la première fois, les mises à jour de sécurité critiques sont installées par défaut. D'autres mises à jour ne sont pas installées. En fonction de la sécurité de votre application et de la durée pendant laquelle un cluster s'exécute, vous pouvez choisir de façon périodique de redémarrer votre cluster pour appliquer des mises à jour de sécurité, ou créer une action d'amorçage pour personnaliser les packages à installer et les mises à jour. Vous pouvez également choisir de tester et d'installer certaines mises à jour de sécurité sur les instances de cluster en cours d'exécution. Pour de plus amples informations, veuillez consulter À l'aide de l'AMI Amazon Linux par défaut pour Amazon EMR (p. 117).

Utilisation de configurations de sécurité pour configurer la sécurité du cluster

Avec Amazon EMR version 4.8.0 ou ultérieure, vous pouvez utiliser des configurations de sécurité pour configurer le chiffrement des données, l'authentification Kerberos (disponible dans la version 5.10.0 et les versions ultérieures) et l'autorisation d'accès à Amazon S3 pour EMRFS (disponible dans la version 5.10.0 et les versions ultérieures).

Une fois que vous avez créé une configuration de sécurité, vous la spécifiez lorsque vous créez un cluster et vous pouvez la réutiliser pour n'importe quel nombre de clusters.

Vous pouvez utiliser la console, l'AWS Command Line Interface (AWS CLI) ou les kits SDK AWS pour créer des configurations de sécurité. Vous pouvez également utiliser un modèle AWS CloudFormation pour créer une configuration de sécurité. Pour plus d'informations, consultez AWS CloudFormation Guide de l'utilisateur et le modèle de référence pour AWS::EMR::SecurityConfiguration.

Rubriques

- Création d'une configuration de sécurité (p. 179)
- Spécification d'une configuration de sécurité pour un cluster (p. 196)

Création d'une configuration de sécurité

Cette rubrique couvre les procédures générales pour la création d'une configuration de sécurité à l'aide de la console EMR et de l'AWS CLI. Elle comporte également une référence pour les paramètres qui comprennent le chiffrement, l'authentification et les rôles IAM pour EMRFS. Pour plus d'informations sur ces fonctions, consultez les rubriques suivantes :

- Chiffrer les données au repos et en transit (p. 197)
- Use Kerberos Authentication (p. 255)
- Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234)

Pour créer une configuration de sécurité à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Dans le volet de navigation, choisissez Security Configurations (Configurations de sécurité), puis Create security configuration (Créer une configuration de sécurité).
3. Dans Name (Nom), saisissez un nom pour la configuration de sécurité.
4. Choisir les options pour Cryptage et Authentification comme décrit dans les sections ci-dessous, puis choisissez Créer.

Pour créer une configuration de sécurité à l'aide de l'AWS CLI

- Utilisez la commande `create-security-configuration`, comme illustré dans l'exemple suivant.
  - Pour : *`SecConfigName`*, spécifiez le nom de la configuration de sécurité. Il s'agit du nom que vous spécifiez lors de la création d'un cluster qui utilise cette configuration de sécurité.
  - Pour *`SecConfigDef`*, spécifiez une structure JSON en ligne ou le chemin vers un fichier JSON local, comme *`file://MySecConfig.json`*. Les paramètres JSON définissent les options pour Cryptage, Rôles IAM pour l'accès à l'EMRFS Amazon S3, et Authentification comme décrit dans les sections ci-dessous.

```
aws emr create-security-configuration --name "SecConfigName" --security-
configuration SecConfigDef
```

Configuration du chiffrement des données

Avant de configurer le chiffrement dans une configuration de sécurité, créez les clés et certificats utilisés pour le chiffrement. Pour de plus amples informations, veuillez consulter Fournir les clés de chiffrement des données au repos avec Amazon EMR (p. 203) et Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR (p. 205).

Lorsque vous créez une configuration de sécurité, vous spécifiez deux jeux d'options de chiffrement : le chiffrement des données au repos et le chiffrement des données en transit. Les options pour un chiffrement de données au repos incluent à la fois Amazon S3 avec EMRFS et le chiffrement de disque local. Les options de chiffrement en transit activent les fonctions de chiffrement open source pour certaines applications qui prennent en charge le protocole TLS (Transport Layer Security). Les options de chiffrement des données au repos et en transit peuvent être activées ensemble ou séparément. Pour plus d'informations, consultez Chiffrer les données au repos et en transit (p. 197),

> Note
>
> Lorsque vous utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour de plus amples informations, veuillez consulter Tarification AWS KMS.

Spécification d'options de chiffrement à l'aide de la console

Choisissez les options sous Encryption (Chiffrement) en fonction des indications suivantes.

- Choisissez les options sous At rest encryption (Chiffrement au repos) pour chiffrer les données stockées dans le système de fichiers.

  Vous pouvez choisir de chiffrer les données dans Amazon S3, les disques locaux ou les deux.
- Sous S3 data encryption (Chiffrement des données S3), pour Encryption mode (Mode de chiffrement), choisissez une valeur qui détermine comment Amazon EMR chiffre les données Amazon S3 avec EMRFS.

  L'étape suivante varie selon le mode de chiffrement que vous avez choisi :
  - SSE-S3

    Spécifie le chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3. Aucune autre action n'est requise de votre part, car Amazon S3 gère les clés pour vous.
  - SSE-KMS ou CSE-KMS

    Spécifie le chiffrement côté serveur avec des clés gérées par AWS KMS (KMS-CSE) ou le chiffrement côté client avec des clés gérées par AWS KMS (CSE-KMS). Pour AWS KMS Key (Clé KMS AWS), sélectionnez une clé. Cette clé doit être dans la même région que votre cluster EMR. Pour voir les exigences relatives aux clés, consultez Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement (p. 203).
  - CSE-Custom

    Spécifie le chiffrement côté client via une clé principale personnalisée côté client (CSE-Custom). Pour S3 object (Objet S3), saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Puis, pour Key provider class (Classe de fournisseur de clé), saisissez le nom complet d'une classe déclarée dans votre application et qui implémente l'interface EncryptionMaterialsProvider.

- Sous Local disk encryption (Chiffrement de disque local), choisissez une valeur pour Key provider type (Type de fournisseur de clé).
  - AWS KMS

    Sélectionnez cette option pour spécifier une clé principale de client (CMK) AWS KMS. Pour AWS KMS customer master key (Clé CMK AWS KMS), sélectionnez une clé. Cette clé doit être dans la même région que votre cluster EMR. Pour plus d'informations sur les exigences relatives aux clés, consultez Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement (p. 203).

    Chiffrement EBS

    Lorsque vous spécifiez AWS KMS comme fournisseur de clés, vous pouvez activer le chiffrement EBS pour chiffrer le périphérique racine et les volumes de stockage EBS. Pour activer cette option, vous devez accorder au rôle de service EMR `EMR_DefaultRole` les autorisations nécessaires pour utiliser la clé CMK que vous spécifiez. Pour plus d'informations sur les exigences relatives aux clés, consultez Activation du chiffrement EBS en fournissant des autorisations supplémentaires pour les clés CMK AWS KMS (p. 204).
  - personnalisé.

    Sélectionnez cette option pour spécifier un fournisseur de clés personnalisé. Pour S3 object (Objet S3), saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Pour Key provider class (Classe de fournisseur de clé), saisissez le nom complet d'une classe déclarée dans votre application et qui implémente l'interface EncryptionMaterialsProvider. Le nom de classe que vous indiquez ici doit être différent du nom de classe fourni pour CSE-Custom.
- Choisissez In-transit encryption (Chiffrement en transit) pour activer les fonctionnalités de chiffrement TLS open source pour les données en transit. Dans Certificate provider type (Type de fournisseur de certificat), sélectionnez un type de fournisseur de certificat conformément aux consignes suivantes :
  - PEM

    Sélectionnez cette option pour utiliser les fichiers PEM que vous fournissez au sein d'un fichier zip. Deux objets sont obligatoires dans le fichier zip : privateKey.pem et certificateChain.pem. Un troisième fichier, trustedCertificates.pem, est facultatif. Consultez Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR (p. 205) pour plus de détails. Pour S3 object (Objet S3), spécifiez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du champ du fichier zip.
  - personnalisé.

    Sélectionnez cette option pour spécifier un fournisseur de certificats personnalisés, puis pour S3 object (Objet S3), saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Pour Key provider class (Classe de fournisseur de clé), saisissez le nom complet d'une classe déclarée dans votre application et qui implémente l'interface TLSArtifactsProvider.

Spécification d'options de chiffrement à l'aide de l'AWS CLI

Les sections suivantes utilisent des exemples de scénarios pour illustrer le code JSON --security-configuration bien formé pour différentes configurations et différents fournisseurs de clés, suivis d'une référence pour les paramètres JSON et les valeurs à utiliser.

Exemple d'options de chiffrement des données en transit

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est activé et le chiffrement des données au repos est désactivé.
- Un fichier zip contenant les certificats dans Amazon S3 est utilisé en tant que fournisseur de clés (voir Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR (p. 205) pour les exigences relatives aux certificats).

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : true,
  "EnableAtRestEncryption" : false,
  "InTransitEncryptionConfiguration" : {
   "TLSCertificateConfiguration" : {
    "CertificateProviderType" : "PEM",
    "S3Object" : "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  }
 }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est activé et le chiffrement des données au repos est désactivé.
- Un fournisseur de clés personnalisé est utilisé (voir Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR (p. 205) pour les exigences relatives aux certificats).

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : true,
  "EnableAtRestEncryption" : false,
  "InTransitEncryptionConfiguration" : {
   "TLSCertificateConfiguration" : {
    "CertificateProviderType" : "Custom",
    "S3Object" : "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass" : "com.mycompany.MyCertProvider"
   }
  }
  }
}'
```

Exemple d'options de chiffrement des données au repos

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est désactivé et le chiffrement des données au repos est activé.
- SSE-S3 est utilisé pour le chiffrement Amazon S3.
- Le chiffrement de disque local utilise AWS KMS comme fournisseur de clés.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : false,
  "EnableAtRestEncryption" : true,
  "AtRestEncryptionConfiguration" : {
   "S3EncryptionConfiguration" : {
    "EncryptionMode" : "SSE-S3"
   },
   "LocalDiskEncryptionConfiguration" : {
    "EncryptionKeyProviderType" : "AwsKms",
    "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
```

```
  }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est activé et fait référence à un fichier zip contenant des certificats PEM dans Amazon S3, à l'aide de l'ARN.
- SSE-KMS est utilisé pour le chiffrement Amazon S3.
- Le chiffrement de disque local utilise AWS KMS comme fournisseur de clés.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : true,
  "EnableAtRestEncryption" : true,
  "InTransitEncryptionConfiguration" : {
   "TLSCertificateConfiguration" : {
    "CertificateProviderType" : "PEM",
    "S3Object" : "arn:aws:s3:::MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration" : {
   "S3EncryptionConfiguration" : {
    "EncryptionMode" : "SSE-KMS",
    "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   },
   "LocalDiskEncryptionConfiguration" : {
    "EncryptionKeyProviderType" : "AwsKms",
    "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
 }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est activé et fait référence à un fichier zip contenant des certificats PEM dans Amazon S3.
- CSE-KMS est utilisé pour le chiffrement Amazon S3.
- Le chiffrement de disque local utilise un fournisseur de clés personnalisé référencé par son ARN.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : true,
  "EnableAtRestEncryption" : true,
  "InTransitEncryptionConfiguration" : {
   "TLSCertificateConfiguration" : {
    "CertificateProviderType" : "PEM",
    "S3Object" : "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration" : {
   "S3EncryptionConfiguration" : {
    "EncryptionMode" : "CSE-KMS",
    "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
```

```
    },
    "LocalDiskEncryptionConfiguration" : {
     "EncryptionKeyProviderType" : "Custom",
     "S3Object" : "arn:aws:s3:::artifacts/MyKeyProvider.jar",
     "EncryptionKeyProviderClass" : "com.mycompany.MyKeyProvider.jar"
    }
   }
  }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est activé avec un fournisseur de clés personnalisé.
- CSE-Custom est utilisé pour les données Amazon S3.
- Le chiffrement de disque local utilise un fournisseur de clés personnalisé.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption" : "true",
  "EnableAtRestEncryption" : "true",
  "InTransitEncryptionConfiguration" : {
   "TLSCertificateConfiguration" : {
    "CertificateProviderType" : "Custom",
    "S3Object" : "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass" : "com.mycompany.MyCertProvider"
   }
  },
  "AtRestEncryptionConfiguration" : {
   "S3EncryptionConfiguration" : {
    "EncryptionMode" : "CSE-Custom",
    "S3Object" : "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass" : "com.mycompany.MyKeyProvider"
    },
   "LocalDiskEncryptionConfiguration" : {
    "EncryptionKeyProviderType" : "Custom",
    "S3Object" : "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass" : "com.mycompany.MyKeyProvider"
   }
  }
 }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est désactivé et le chiffrement des données au repos est activé.
- Le chiffrement Amazon S3 est activé avec SSE-KMS et les exceptions de chiffrement sont appliquées aux compartiments S3 individuels.
- Le chiffrement de disque local est désactivé.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
   "EncryptionConfiguration": {
     "AtRestEncryptionConfiguration": {
             "S3EncryptionConfiguration": {
           "EncryptionMode" : "SSE-KMS",
           "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012",
           "Overrides" : [
```

```
              {
                "BucketName" : "sse-s3-bucket-name",
                "EncryptionMode" : "SSE-S3"
              },
              {
                "BucketName" : "cse-kms-bucket-name",
                "EncryptionMode" : "CSE-KMS",
                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              },
              {
                "BucketName" : "sse-kms-bucket-name",
                "EncryptionMode" : "SSE-KMS",
                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              }
             ]
             }
           },
     "EnableInTransitEncryption": false,
      "EnableAtRestEncryption": true
  }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est désactivé et le chiffrement des données au repos est activé.
- Le chiffrement Amazon S3 est activé avec SSE-S3 et le chiffrement de disque local est désactivé.

```
aws emr create-security-configuration --name "MyS3EncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption" : false,
        "EnableAtRestEncryption" : true,
        "AtRestEncryptionConfiguration" : {
            "S3EncryptionConfiguration" : {
                "EncryptionMode" : "SSE-S3"
            }
        }
     }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est désactivé et le chiffrement des données au repos est activé.
- Le chiffrement de disque local est activé avec AWS KMS comme fournisseur de clés et le chiffrement Amazon S3 est désactivé.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption" : false,
        "EnableAtRestEncryption" : true,
        "AtRestEncryptionConfiguration" : {
            "LocalDiskEncryptionConfiguration" : {
                "EncryptionKeyProviderType" : "AwsKms",
                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
```

```
    }
}'
```

L'exemple suivant illustre le scénario suivant :

- Le chiffrement des données en transit est désactivé et le chiffrement des données au repos est activé.
- Le chiffrement de disque local est activé avec AWS KMS comme fournisseur de clés et le chiffrement Amazon S3 est désactivé.
- Le chiffrement EBS est activé.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption" : false,
        "EnableAtRestEncryption" : true,
        "AtRestEncryptionConfiguration" : {
            "LocalDiskEncryptionConfiguration" : {
                "EnableEbsEncryption" : true,
                "EncryptionKeyProviderType" : "AwsKms",
                "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}'
```

Référence JSON pour les paramètres de chiffrement

Le tableau suivant répertorie les paramètres JSON à définir pour le chiffrement et décrit les valeurs acceptables pour chacun d'eux.

Paramètre :	Description :
`"EnableInTransitEncryption" : true | false`	Spécifiez `true` pour activer le chiffrement en transit et `false` pour le désactiver. S'il est omis, `false` est utilisé par défaut, et le chiffrement en transit est désactivé.
`"EnableAtRestEncryption" : true | false`	Spécifiez `true` pour activer le chiffrement au repos et `false` pour le désactiver. S'il est omis, `false` est utilisé par défaut, et le chiffrement au repos est désactivé.
Paramètres de chiffrement en transit	
`"InTransitEncryptionConfiguration" :`	Spécifie une collection de valeurs utilisées pour configurer le chiffrement en transit quand `EnableInTransitEncryption` est `true`.
`"CertificateProviderType" : "PEM" | "Custom"`	Spécifie si les certificats `PEM` référencés avec un fichier zip ou un fournisseur de certificats `Custom` doivent être utilisés. Si `PEM` est défini, `S3Object` doit renvoyer à l'emplacement dans Amazon S3 d'un fichier zip contenant les certificats. Si Custom est spécifié, `S3Object` doit renvoyer à l'emplacement dans Amazon S3 d'un fichier JAR, suivi d'une entrée `CertificateProviderClass`.

Paramètre :	Description :
`"S3Object" : "ZipLocation" | "JarLocation"`	Fournit l'emplacement dans Amazon S3 d'un fichier zip quand `PEM` est spécifié, ou d'un fichier JAR quand `Custom` est spécifié. Le format peut être un chemin (par exemple, `s3://MyConfig/artifacts/CertFiles.zip`) ou un ARN (par exemple, `arn:aws:s3:::Code/MyCertProvider.jar`). Si un fichier zip est spécifié, il doit contenir des fichiers nommés `privateKey.pem` et `certificateChain.pem`. Un fichier nommé `trustedCertificates.pem` est facultatif.
`"CertificateProviderClass" : "MyClassID"`	Nécessaire seulement si `Custom` est spécifié pour `CertificateProviderType`. *`MyClassID`* spécifie le nom complet d'une classe déclarée dans le fichier JAR et qui implémente l'interface TLSArtifactsProvider. Par exemple : `com.mycompany.MyCertProvider`.
Paramètres de chiffrement au repos	
`"AtRestEncryptionConfiguration" :`	Spécifie une collection de valeurs pour le chiffrement au repos quand `EnableAtRestEncryption` est `true`, y compris le chiffrement Amazon S3 et le chiffrement de disque local.
Amazon S3 paramètres de chiffrement	
`"S3EncryptionConfiguration" :`	Spécifie une collection de valeurs utilisées pour le chiffrement Amazon S3 avec le système de fichiers EMR (EMRFS).
`"EncryptionMode" : "SSE-S3" | "SSE-KMS" | "CSE-KMS" | "CSE-Custom"`	Spécifie le type de chiffrement Amazon S3 à utiliser. Si `SSE-S3` est spécifié, aucune autre valeur de chiffrement Amazon S3 n'est requise. Si `SSE-KMS` ou `CSE-KMS` est défini, l'ARN d'une clé principale de client (CMK) AWS KMS doit être spécifiée comme valeur `AwsKmsKey`. Si `CSE-Custom` est défini, les valeurs `S3Object` et `EncryptionKeyProviderClass` doivent être spécifiées.
`"AwsKmsKey" : "MyKeyARN"`	Obligatoire uniquement quand `SSE-KMS` ou `CSE-KMS` est spécifié pour `EncryptionMode`. *`MyKeyARN`* doit être l'ARN complet d'une clé (par exemple, `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-1`
`"S3Object" : "JarLocation"`	Obligatoire uniquement quand `CSE-Custom` est spécifié pour `CertificateProviderType`. *`JarLocation`* fournit l'emplacement dans Amazon S3 à un fichier JAR. Le format peut être un chemin d'accès (par exemple, `s3://MyConfig/artifacts/MyKeyProvider.jar`) ou un ARN (par exemple, `arn:aws:s3:::Code/MyKeyProvider.jar`).

Paramètre :	Description :
`"EncryptionKeyProviderClass" : "MyS3KeyClassID"`	Obligatoire uniquement quand `CSE-Custom` est spécifié pour `EncryptionMode`. *`MyS3KeyClassID`* spécifie le nom complet d'une classe déclarée dans l'application et qui implémente l'interface EncryptionMaterialsProvider (par exemple, *`com.mycompany.MyS3KeyProvider`*).
Paramètres de chiffrement de disque local	
`"LocalDiskEncryptionConfiguration"`	Spécifie le fournisseur de clés et les valeurs correspondantes à utiliser pour le chiffrement de disque local.
`"EncryptionKeyProviderType" : "AwsKms" | "Custom"`	Spécifie le fournisseur de clés. Si `AwsKms` est défini, l'ARN de la clé CMK AWS KMS doit être spécifiée en tant que valeur `AwsKmsKey`. Si **`Custom`** est défini, les valeurs `S3Object` et `EncryptionKeyProviderClass` doivent être spécifiées.
`"AwsKmsKey : "MyKeyARN"`	Obligatoire uniquement quand `AwsKms` est spécifié pour `Type`. *`MyKeyARN`* doit être l'ARN complet d'une clé (par exemple, `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-4`
`"S3Object" : "JarLocation"`	Obligatoire uniquement quand `CSE-Custom` est spécifié pour `CertificateProviderType`. *`JarLocation`* fournit l'emplacement dans Amazon S3 à un fichier JAR. Le format peut être un chemin d'accès (par exemple, `s3://MyConfig/artifacts/MyKeyProvider.jar`) ou un ARN (par exemple, `arn:aws:s3:::Code/MyKeyProvider.jar`).
`"EncryptionKeyProviderClass" : "MyLocalDiskKeyClassID"`	Obligatoire uniquement quand `Custom` est spécifié pour `Type`. *`MyLocalDiskKeyClassID`* spécifie le nom complet d'une classe déclarée dans l'application et qui implémente l'interface EncryptionMaterialsProvider (par exemple, *`com.mycompany.MyLocalDiskKeyProvider`*).

Configuration de l'authentification Kerberos

Une configuration de sécurité avec les paramètres Kerberos ne peut être utilisée que par un cluster créé avec des attributs Kerberos. Sinon, une erreur se produit. Pour plus d'informations, consultez Use Kerberos Authentication (p. 255), Kerberos est uniquement disponible dans Amazon EMR version 5.10.0 et versions ultérieures.

Spécification des paramètres Kerberos à l'aide de la console

Choisissez les options sous Kerberos authentication (Authentification Kerberos) en suivant les indications suivantes.

Paramètre		Description
Kerberos		Spécifie que Kerberos est activé pour les clusters qui utilisent cette configuration de sécurité. Si un cluster utilise cette configuration de sécurité, les paramètres Kerberos doivent également être spécifiés pour celui-ci, sinon, une erreur se produit.
Fournisseur	KDC dédié au cluster	Spécifie que Amazon EMR crée un KDC sur le nœud principal de tout cluster qui utilise cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administrateur KDC au moment de créer le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaires. Créez ces clusters en utilisant une configuration de sécurité différente, spécifiez un KDC externe et utilisez le nom de domaine et le mot de passe administrateur KDC que vous avez spécifiés pour le KDC dédié au cluster.
	KDC externe	Disponible uniquement avec Amazon EMR 5.20.0 et versions ultérieures. Indique que les clusters qui utilisent cette configuration de sécurité authentifient les mandataires Kerberos à l'aide d'un serveur KDC externe au cluster. Aucun KDC n'est créé sur le cluster. Lorsque vous créez un cluster, vous spécifiez le nom de domaine et le mot de passe administrateur KDC pour le KDC externe.
Durée de vie du ticket		Facultatif. Spécifie la période pour laquelle un ticket Kerberos émis par le KDC est valide sur les clusters qui utilisent cette configuration de sécurité. La durée de vie des tickets est limitée pour des raisons de sécurité. Les applications et services de cluster renouvellent automatiquement les tickets après leur expiration. Les utilisateurs qui se connectent au cluster via SSH à l'aide d'informations d'identification Kerberos doivent exécuter `kinit` à partir de la ligne de commande du nœud principal pour renouveler un ticket après son expiration.
Relation d'approbation inter-domaines		Spécifie une relation d'approbation inter-domaines entre un KDC dédié au cluster sur les clusters qui utilisent cette configuration de sécurité et un KDC situé dans un autre domaine Kerberos. Les mandataires (généralement des utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est nécessaire. Pour plus d'informations, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277).
Propriétés de la relation d'approbation inter-domaines	Domaine	Spécifie le nom de domaine Kerberos de l'autre domaine inclus dans la relation d'approbation. Par convention, les noms de domaine Kerberos sont identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.

Paramètre		Description
	Domaine	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.
	Serveur d'administration	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:749`).
	Serveur KDC	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur KDC de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:88`).
KDC externe		Indique que le KDC externe de cluster est utilisé par le cluster.
Propriétés du KDC externe	Serveur d'administration	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:749`).
	Serveur KDC	Indique le nom de domaine complet (FQDN) du serveur KDC externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:88`).
	Intégration d'Active Directory	Indique que l'authentification du mandataire Kerberos est intégrée à un domaine Microsoft Active Directory.

Paramètre			Description
	Propriétés d'intégration d'Active Directory	Active Directory realm (Domaine (realm) Active Directory)	Indique le nom du domaine (realm) Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.
		Active Directory domain (Domaine Active Directory)	Indique le nom de domaine Active Directory.
		Active Directory server (Serveur Active Directory)	Indique le nom de domaine complet (FQDN) du contrôleur de domaine Microsoft Active Directory.

Spécification des paramètres Kerberos à l'aide de l'AWS CLI

Le tableau suivant montre les paramètres JSON de référence pour les paramètres Kerberos dans une configuration de sécurité. Pour des exemples de configurations, consultez Configuration Examples (p. 270).

Paramètre			Description
`"AuthenticationConfiguration": {`			Nécessaire pour Kerberos. Indique qu'une configuration d'authentification fait partie de cette configuration de sécurité.
	`"KerberosConfiguration": {`		Nécessaire pour Kerberos. Spécifie les propriétés de la configuration Kerberos.

		`"Provider": "ClusterDedicatedKdc",`
		—ou—
		`"Provider: "ExternalKdc",`
			`ClusterDedicatedKdc` indique qu'Amazon EMR crée un KDC sur le nœud principal de tout cluster qui utilise cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administrateur KDC au moment de créer le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaires. Créez ces clusters en utilisant une configuration de sécurité différente, spécifiez un KDC externe et utilisez le nom de domaine et le mot de passe administrateur KDC que vous avez spécifiés au moment de créer le cluster avec le KDC dédié au cluster.

Paramètre	Description
	ExternalKdc indique que le cluster utilise un KDC externe. Amazon EMR ne crée pas de KDC sur le nœud principal. Un cluster qui utilise cette configuration de sécurité doit spécifier le nom de domaine et le mot de passe administrateur du KDC externe.
`"ClusterDedicatedKdcConfiguration": {`	Obligatoire lorsque *ClusterDedicatedKdc* est spécifié.
`"TicketLifetimeInHours": 24,`	Facultatif. Spécifie la période pour laquelle un ticket Kerberos émis par le KDC est valide sur les clusters qui utilisent cette configuration de sécurité. La durée de vie des tickets est limitée pour des raisons de sécurité. Les applications et services de cluster renouvellent automatiquement les tickets après leur expiration. Les utilisateurs qui se connectent au cluster via SSH à l'aide d'informations d'identification Kerberos doivent exécuter `kinit` à partir de la ligne de commande du nœud principal pour renouveler un ticket après son expiration.
`"CrossRealmTrustConfiguration": {`	Spécifie une relation d'approbation inter-domaines entre un KDC dédié au cluster sur les clusters qui utilisent cette configuration de sécurité et un KDC situé dans un autre domaine Kerberos. Les mandataires (généralement des utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est nécessaire. Pour plus d'informations, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277).
`"Realm": "KDC2.COM",`	Spécifie le nom de domaine Kerberos de l'autre domaine inclus dans la relation d'approbation. Par convention, les noms de domaine Kerberos sont identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.
`"Domain": "kdc2.com",`	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.

Paramètre	Description
`"AdminServer": "kdc.com:749",`	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:749`).
`"KdcServer": "kdc.com:88"`	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur KDC de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:88`).
`}`	
`}`	
"ExternalKdcConfiguration": {	Obligatoire lorsque *`ExternalKdc`* est spécifié.
`"TicketLifetimeInHours": 24,`	Facultatif. Spécifie la période pour laquelle un ticket Kerberos émis par le KDC est valide sur les clusters qui utilisent cette configuration de sécurité. La durée de vie des tickets est limitée pour des raisons de sécurité. Les applications et services de cluster renouvellent automatiquement les tickets après leur expiration. Les utilisateurs qui se connectent au cluster via SSH à l'aide d'informations d'identification Kerberos doivent exécuter `kinit` à partir de la ligne de commande du nœud principal pour renouveler un ticket après son expiration.

Paramètre	Description
`"KdcServerType": "Single",`	Indique qu'un serveur KDC est référencé. `Single` est actuellement la seule valeur prise en charge.
"AdminServer": "*`kdc.com:749`*",	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:`*`749`*).
"KdcServer": "*`kdc.com:88`*",	Indique le nom de domaine complet (FQDN) du serveur KDC externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:`*`88`*).
`"AdIntegrationConfiguration": {`	Indique que l'authentification du mandataire Kerberos est intégrée à un domaine Microsoft Active Directory.
`"AdRealm": "`*`AD.DOMAIN.COM`*`",`	Indique le nom du domaine (realm) Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.
`"AdDomain": "`*`ad.domain.com`*`"`	Indique le nom de domaine Active Directory.
`}`	
`}`	
`}`	
`}`	

Configuration de rôles IAM pour les demandes EMRFS à Amazon S3

Les rôles IAM pour EMRFS vous permettent de fournir différentes autorisations pour les données EMRFS dans Amazon S3. Vous créez des mappages qui spécifient un rôle IAM utilisé pour les autorisations lorsqu'une demande d'accès contient un identificateur que vous spécifiez. L'identificateur peut être un utilisateur ou un rôle Hadoop, ou bien un préfixe Amazon S3.

Pour plus d'informations, consultez Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234),

Spécification de rôles IAM pour EMRFS à l'aide de l'AWS CLI.

Voici un exemple d'extrait JSON permettant de spécifier des rôles IAM personnalisés pour EMRFS dans une configuration de sécurité. Cet exemple illustre des mappages de rôles pour les trois différents types d'identifiant, suivis par une référence de paramètre.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Paramètre	Description
`"AuthorizationConfiguration":`	Obligatoire.
`"EmrFsConfiguration":`	Obligatoire. Contient les mappages de rôle.
`"RoleMappings":`	Obligatoire. Contient une ou plusieurs définitions de mappage de rôle. Les mappages de rôle sont évalués dans l'ordre descendant dans lequel ils s'affichent. Si un mappage de rôle est évalué comme true pour un appel EMRFS de données dans Amazon S3, aucun autre mappage de rôle n'est évalué et EMRFS utilise le rôle IAM spécifié pour la demande. Les mappages de rôle comprennent les paramètres obligatoires suivants :
`"Role":`	Spécifie l'identifiant ARN d'un rôle IAM au format `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*. Il s'agit du rôle IAM assumé par Amazon EMR si la demande EMRFS à Amazon S3 correspond à l'un des `Identifiers` spécifiés.
`"IdentifierType":`	Peut avoir l'une des valeurs suivantes :

Paramètre	Description
	• `"User"` spécifie que les identifiants correspondent à un ou plusieurs utilisateurs Hadoop qui peuvent être des utilisateurs de compte Linux ou des principaux Kerberos. Lorsque la demande EMRFS provient du ou des utilisateurs spécifiés, le rôle IAM est assumé. • `"Prefix"` indique que l'identifiant correspond à un emplacement Amazon S3. Le rôle IAM est assumé pour les appels vers le ou les emplacements avec les préfixes spécifiés. Par exemple, le préfixe `s3://mybucket/` correspond à `s3://mybucket/mydir` et à `s3://mybucket/yetanotherdir`. • `"Group"` indique que les identifiants correspondent à un ou plusieurs groupes Hadoop. Le rôle IAM est assumé lorsque la demande provient d'un utilisateur du ou des groupes spécifiés.
`"Identifiers":`	Spécifie un ou plusieurs identifiants du type approprié. Séparez les identifiants par des virgules sans espace.

Spécification d'une configuration de sécurité pour un cluster

Vous pouvez spécifier les paramètres de chiffrement lorsque vous créez un cluster en spécifiant la configuration de sécurité. Pour ce faire, vous pouvez utiliser la AWS Management Console ou l'AWS CLI.

Spécification d'une configuration de sécurité à l'aide de la console

Lorsque vous utilisez le AWS pour créer un cluster EMR, vous choisissez la configuration de sécurité pendant Étape 4 : Sécurité du processus de création des options avancées.

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Créer un cluster et Go to advanced options (Aller aux options avancées).
3. Sur leÉtape 1 : Logiciels et étapes à partir du Libération liste, choisissez emr-4.8.0 ou une version plus récente. Choisissez les paramètres que vous voulez, puis cliquez sur Next (Suivant).
4. Sur le Étape 2 : Matériel choisissez les paramètres souhaités et choisissez Suivant. Faites de même pour Étape 3 : Paramètres généraux du cluster.
5. Sur le Étape 4 : Sécurité écran, sous Options de chiffrement, choisissez une valeur pour Configuration de sécurité.
6. Configurez les autres options de sécurité comme vous le souhaitez et choisissez Create cluster (Créer le cluster).

Spécification d'une configuration de sécurité à l'aide de l'interface de ligne de commande

Lorsque vous utilisez `aws emr create-cluster`, vous pouvez appliquer le cas échéant une configuration de sécurité avec `--security-configuration` *`MySecConfig`*, où *`MySecConfig`*

correspond au nom de la configuration de sécurité, comme illustré dans l'exemple suivant. La valeur `--release-label` spécifiée doit être supérieure ou égale à 4.8.0 et le paramètre `--instance-type` peut être m'importe quel type d'instance disponible.

```
aws emr create-cluster --instance-type m5.xlarge --release-label emr-5.0.0 --security-
configuration mySecConfig
```

Protection des données dans Amazon EMR

Amazon EMR se conforme au modèle de responsabilité partagée AWS, qui inclut des directives et des régulations pour la protection des données. AWS est responsable de la protection de l'infrastructure globale qui exécute tous les services AWS. AWS conserve le contrôle des données hébergées sur cette infrastructure, y compris les contrôles de configuration de sécurité pour traiter le contenu et les données personnelles des clients. Les clients AWS et les partenaires d'APN, qu'ils agissent en tant que contrôleurs ou responsables du traitement des données, sont responsables de toutes les données personnelles qu'ils mettent sur le cloud AWS.

Pour des raisons de protection des données, nous vous recommandons de protéger vos identifiants de compte AWS et de configurer des comptes utilisateurs individuels avec AWS Identity and Access Management (IAM), afin que chaque utilisateur reçoive uniquement les permissions nécessaires pour accomplir ses tâches. Nous vous recommandons également de sécuriser vos données de la manière suivante :

- Utilisez les options de chiffrement Amazon EMR pour chiffrer les données au repos et en transit. Pour plus d'informations, consultez Chiffrer les données au repos et en transit (p. 197),
- Utilisation de l'authentification multi-facteurs (MFA) avec chaque compte.
- Utilisez SSL/TLS pour communiquer avec des ressources AWS.
- Configurez l'API et la journalisation des activités utilisateur avec AWS CloudTrail.
- Utilisez des services de sécurité gérés avancés comme Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.

Nous vous recommandons vivement de ne jamais placer d'informations identifiables sensibles, telles que les numéros de compte de vos clients, dans des champs de formulaire comme Name (Nom). Cela s'applique aussi lorsque vous utilisez Amazon EMR ou d'autres services AWS à l'aide de la console, de l'API, de l'AWS CLI ou des kits SDK AWS. Toutes les données que vous entrez dans Amazon EMR ou d'autres services peuvent être récupérées, afin d'être insérées dans des journaux de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Pour de plus amples informations sur la protection des données, veuillez consulter le billet de blog Modèle de responsabilité partagée AWS et RGPD sur le Blog sur la sécurité d'AWS.

Chiffrer les données au repos et en transit

Le chiffrement des données vous permet d'empêcher les utilisateurs non autorisés de lire les données d'un cluster et celles des systèmes de stockage de données associés. Cela inclut les données enregistrées sur les supports persistants (données au repos) et les données qui peuvent être interceptées alors qu'elles circulent sur le réseau (données en transit).

A partir de la version 4.8.0 d'Amazon EMR, vous pouvez utiliser les configurations de sécurité Amazon EMR pour configurer plus facilement les paramètres de chiffrement de données pour les clusters. Les

configurations de sécurité offrent des paramètres permettant de garantir la sécurité des données en transit et des données au repos dans Amazon Elastic Block Store (Amazon EBS) volumes et EMRFS sur Amazon S3.

Le cas échéant, à partir des versions 4.1.0 et ultérieures d'Amazon EMR, vous pouvez choisir de configurer un chiffrement transparent dans HDFS, qui n'est pas configuré à l'aide de configurations de sécurité. Pour plus d'informations, consultez Chiffrement transparent dans HDFS sur Amazon EMR dans le Amazon EMR Guide de version.

Rubriques

- Options de chiffrement (p. 198)
- Création de clés et de certificats pour le chiffrement des données (p. 203)

Options de chiffrement

Avec la version 4.8.0 d'Amazon EMR, vous pouvez utiliser une configuration de sécurité pour spécifier les paramètres de chiffrement des données au repos, des données en transit ou des deux. Lorsque vous activez le chiffrement des données au repos, vous pouvez choisir de chiffrer les données EMRFS dans Amazon S3, les données dans les disques locaux, ou les deux. Chaque configuration de sécurité créée est stockée dans Amazon EMR plutôt que dans la configuration du cluster. Dès lors, vous pouvez facilement réutiliser une configuration pour spécifier les paramètres de chiffrement de données chaque fois qu'un cluster est créé. Pour plus d'informations, consultez Création d'une configuration de sécurité (p. 179),

Le schéma suivant illustre les différentes options de chiffrement des données disponibles avec les configurations de sécurité.

Les options de chiffrement suivantes sont également disponibles et ne sont pas configurées à l'aide d'une configuration de sécurité :

- Le cas échéant, avec les versions Amazon EMR 4.1.0 et ultérieures, vous pouvez choisir de configurer le chiffrement transparent dans HDFS. Pour plus d'informations, consultez Chiffrement transparent dans HDFS sur Amazon EMR dans le Amazon EMR Guide de version.
- Si vous utilisez une version d'Amazon EMR qui ne prend pas en charge les configurations de sécurité, vous pouvez configurer manuellement le chiffrement pour les données EMRFS dans Amazon S3. Pour plus d'informations, consultez Spécification du chiffrement Amazon S3 en utilisant les propriétés EMRFS (p. 102),
- Si vous utilisez une version Amazon EMR antérieure à 5.24.0, un volume de périphérique racine EBS chiffré est pris en charge uniquement lorsque vous utilisez une AMI personnalisée. Pour plus d'informations, consultez Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré dans le Amazon EMR Guide de gestion.

Note

À partir de Amazon EMR version 5.24.0, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous

spécifiez AWS KMS comme fournisseur de clé. Pour plus d'informations, consultez Chiffrement de disque local (p. 201).

Le chiffrement des données nécessite des clés et des certificats. Une configuration de sécurité vous donne la possibilité de choisir parmi plusieurs options, y compris les clés gérées par AWS Key Management Service, les clés gérées par Amazon S3 et les clés et les certificats provenant de fournisseurs personnalisés que vous fournissez. Lorsque vous utilisez AWS KMS comme fournisseur de clés, des frais vous sont facturés pour le stockage et l'utilisation des clés de chiffrement. Pour de plus amples informations, veuillez consulter Tarification AWS KMS.

Avant de spécifier les options de chiffrement, choisissez les systèmes de gestion de clés et de certificats que vous voulez utiliser et commencez par créer les clés et les certificats ou les fournisseurs personnalisés que vous définissez dans le cadre des paramètres de chiffrement.

Chiffrement au repos pour les données EMRFS dans Amazon S3

Le chiffrement Amazon S3 fonctionne avec les objets du système de fichiers EMR (EMRFS) lus et écris sur Amazon S3. Vous spécifiez le chiffrement côté serveur (SSE) ou côté client (CSE) Amazon S3 comme mode de chiffrement par défaut lorsque vous activez le chiffrement au repos. Le cas échéant, vous pouvez spécifier différentes méthodes de chiffrement pour les compartiments individuels à l'aide de remplacements de chiffrement par compartiment. Que le chiffrement Amazon S3 soit activé ou non, le protocole TLS (Transport Layer Security) chiffre les objets EMRFS en transit entre les nœuds de cluster EMR et Amazon S3. Pour des informations approfondies sur le chiffrement Amazon S3, consultez Protection des données à l'aide d'un chiffrement dans le Amazon Simple Storage Service Manuel du développeur.

> **Note**
>
> Lorsque vous utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour de plus amples informations, veuillez consulter Tarification AWS KMS.

Chiffrement Amazon S3 côté serveur

Lorsque vous configurez le chiffrement Amazon S3 côté serveur, Amazon S3 chiffre les données au niveau de l'objet lors de l'écriture des données sur le disque et les déchiffre lorsque quelqu'un y accède. Pour plus d'informations sur le chiffrement SSE, consultez Protection des données à l'aide d'un chiffrement côté serveur dans le Amazon Simple Storage Service Manuel du développeur.

Vous pouvez choisir entre deux systèmes de gestion des clés différents quand vous spécifiez SSE dans Amazon EMR :

- SSE-S3 – Amazon S3 gère les clés pour vous.
- SSE-KMS – Vous utilisez une clé principale du client (CMK) AWS KMS configurée avec des stratégies adaptées à Amazon EMR. Pour plus d'informations sur les exigences relatives aux clés pour Amazon EMR, consultez Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement (p. 203).

Le chiffrement SSE avec des clés fournies par le client (SSE-C) n'est pas disponible avec Amazon EMR.

Chiffrement Amazon S3 côté client

Avec le chiffrement Amazon S3 côté client, le chiffrement et le déchiffrement Amazon S3 se produisent dans le client EMRFS, sur votre cluster. Les objets sont chiffrés avant d'être chargés dans Amazon S3 et déchiffrés après leur téléchargement. Le fournisseur que vous spécifiez fournit la clé de chiffrement que le client utilise. Le client peut utiliser les clés fournies par AWS KMS (CSE-KMS) ou une classe Java personnalisée qui fournit la clé principale côté client (CSE-C). Les détails de chiffrement sont légèrement différents entre CSE-KMS et CSE-C, selon le fournisseur spécifié et les métadonnées de l'objet déchiffré ou chiffré. Pour de plus amples informations sur ces différences, veuillez consulter Protection des données via le chiffrement côté client dans le Amazon Simple Storage Service Manuel du développeur.

> **Note**
>
> Le chiffrement côté client Amazon S3 garantit uniquement le chiffrement des données EMRFS échangées avec Amazon S3. Les données des volumes d'instance du cluster ne sont pas toutes chiffrées. En outre, comme Hue n'utilise pas EMRFS, les objets écrits sur Amazon S3 à l'aide de l'explorateur de fichiers S3 Hue ne sont pas chiffrés.

Chiffrement de disque local

Les mécanismes suivants fonctionnent ensemble pour chiffrer les disques locaux lorsque vous activez le chiffrement de disque local à l'aide d'une configuration de sécurité Amazon EMR.

Chiffrement HDFS open source

HDFS échange des données entre les instances de cluster pendant le traitement distribué. Il lit et écrit également des données sur les volumes de stockage d'instance et les volumes EBS attachés aux instances. Les options de chiffrement open source Hadoop suivantes sont activées lorsque vous mettez en œuvre le chiffrement de disque local :

- Secure Hadoop RPC est défini sur `Privacy`, qui utilise Simple Authentication Security Layer (SASL).
- Data encryption on HDFS block data transfer est défini sur `true` et est configuré pour utiliser le chiffrement AES 256.

> **Note**
>
> Vous pouvez activer un chiffrement Apache Hadoop supplémentaire en mettant en ouvre le chiffrement en transit (voir Chiffrement en transit (p. 202)). Ces paramètres de chiffrement n'activent pas le chiffrement transparent HDFS, que vous pouvez configurer manuellement. Pour plus d'informations, consultez Chiffrement transparent dans HDFS sur Amazon EMR dans le Amazon EMR Guide de version.

Chiffrement du stockage d'instance

Pour les types d'instance EC2 qui utilisent des disques SSD NVMe comme volume de stockage d'instance, le chiffrement NVMe est utilisé quels que soient les paramètres de chiffrement Amazon EMR. Pour plus d'informations, consultez Volumes SSD NVMe dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Pour les autres volumes de stockage d'instance, Amazon EMR utilise LUKS pour chiffrer le volume de stockage d'instance lorsque le chiffrement de disque local est activé, que les volumes EBS soient chiffrés à l'aide du chiffrement EBS ou LUKS.

Chiffrement de volume EBS

Si vous créez un cluster dans une région où le chiffrement Amazon EC2 des volumes EBS est activé par défaut pour votre compte, les volumes EBS sont chiffrés même si le chiffrement de disque local n'est pas activé. Pour plus d'informations, consultez Chiffrement par défaut dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Avec le chiffrement de disque local activé dans une configuration de sécurité, les paramètres Amazon EMR ont priorité sur les paramètres de chiffrement Amazon EC2 par défaut pour les instances EC2 du cluster.

Les options suivantes sont disponibles pour chiffrer les volumes EBS à l'aide d'une configuration de sécurité :

- Chiffrement EBS – À partir de Amazon EMR version 5.24.0, vous pouvez choisir d'activer le chiffrement EBS. L'option de chiffrement EBS chiffre le volume du périphérique racine EBS et les volumes de stockage attachés. L'option de chiffrement EBS est disponible uniquement lorsque vous spécifiez AWS Key Management Service comme fournisseur de clés. Nous vous recommandons d'utiliser le chiffrement EBS.

- Chiffrement LUKS – Si vous choisissez d'utiliser le chiffrement LUKS pour les volumes Amazon EBS, le chiffrement LUKS s'applique uniquement aux volumes de stockage attachés, pas au volume du périphérique racine. Pour en savoir plus sur le chiffrement LUKS, consultez la spécification de LUKS sur le disque.

  Pour le fournisseur de clés, vous pouvez configurer une clé CMK AWS KMS avec des stratégies stratégies adaptées à Amazon EMR ou une classe Java personnalisée qui fournit les artefacts de chiffrement. Lorsque vous utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour de plus amples informations, veuillez consulter Tarification AWS KMS.

  Note

  Pour vérifier si le chiffrement EBS est activé sur votre cluster, il est recommandé d'utiliser un appel d'API `DescribeVolumes`. Pour plus d'informations, consultez DescribeVolumes. L'exécution de `lsblk` sur le cluster vérifie uniquement le statut de chiffrement LUKS, au lieu du chiffrement EBS.

Chiffrement en transit

Plusieurs mécanismes de chiffrement sont activés avec le chiffrement en transit. Il s'agit de fonctionnalités open source et spécifiques à l'application, qui peuvent varier selon la version Amazon EMR. Les fonctionnalités suivantes de chiffrement propres à l'application peuvent être activées à l'aide de configurations de sécurité :

- Hadoop (pour plus d'informations, consultez Hadoop en mode sécurisé dans la documentation Apache Hadoop) :
  - Hadoop MapReduce Encrypted Shuffle utilise le protocole TLS.
  - RPC Hadoop sécurisé est défini sur « Privacy » (Privé) et utilise SASL (activé dans Amazon EMR lorsque le chiffrement au repos est mis en œuvre).
  - Chiffrement des données sur le transfert de données en bloc HDFS utilise AES 256 (activé dans Amazon EMR lorsque le chiffrement au repos est mis en œuvre dans la configuration de sécurité).
- HBase
  - Lorsque Kerberos est activé, la propriété `hbase.rpc.protection` est définie sur `privacy` pour les communications chiffrées. Pour plus d'informations, consultez Configuration côté-client pour des opérations sécurisées dans la documentation Apache HBase. Pour plus d'informations sur Kerberos avec Amazon EMR, consultez Use Kerberos Authentication (p. 255).
- , Presto
  - Les communications internes entre les nœuds Presto utilisent SSL/TLS (Amazon EMR version 5.6.0 et ultérieures uniquement).
- Tez
  - Gestionnaire Tez Shuffle utilise le protocole TLS (`tez.runtime.ssl.enable`).
- Spark (pour plus d'informations, consultez Paramètres de sécurité Spark) :
  - La communication RPC interne entre les composants Spark, par exemple, le service de transfert de blocs et le service shuffle externe, est chiffrée à l'aide de l'algorithme AES-256 dans Amazon EMR versions 5.9.0 et ultérieures. Dans les versions antérieures, cette communication est chiffrée avec SASL et DIGEST-MD5 (algorithme de chiffrement).
  - Les communications HTTP avec les interfaces utilisateur comme le serveur d'historique Spark et les serveurs de fichiers HTTPS sont chiffrées à l'aide de la configuration SSL de Spark. Pour plus d'informations, consultez Configuration SSL dans la documentation Spark.

Vous spécifiez les objets de chiffrement utilisés pour le chiffrement en transit de l'une des deux façons suivantes : en fournissant un fichier compressé contenant les certificats que vous chargez dans Amazon S3 ou en renvoyant vers une classe Java personnalisée qui fournit les objets de chiffrement. Pour plus

d'informations, consultez Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR (p. 205),

Création de clés et de certificats pour le chiffrement des données

Avant de spécifier les options de chiffrement à l'aide d'une configuration de sécurité, choisissez le fournisseur que vous souhaitez utiliser pour les clés et les artefacts de chiffrement. Par exemple, vous pouvez utiliser AWS KMS ou un fournisseur personnalisé que vous créez. Ensuite, créez les clés ou le fournisseur de clés comme décrit dans cette section.

Fournir les clés de chiffrement des données au repos avec Amazon EMR

Vous pouvez utiliser AWS Key Management Service (AWS KMS) ou un fournisseur de clés personnalisé pour le chiffrement des données au repos dans Amazon EMR. Lorsque vous utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour de plus amples informations, veuillez consulter Tarification AWS KMS.

Cette rubrique fournit des informations sur la stratégie de la clé CMK AWS KMS à utiliser avec Amazon EMR, ainsi que des instructions et des exemples de code pour écrire une classe de fournisseur de clés personnalisé pour le chiffrement Amazon S3. Pour plus d'informations sur la création de clés, consultez Création de clés dans le AWS Key Management Service Developer Guide.

Utilisation des clés principales client (CMK) AWS KMS pour le chiffrement

La clé de chiffrement AWS KMS doit être créée dans la même région que votre instance de cluster Amazon EMR et les compartiments Amazon S3 utilisés avec EMRFS. Si la clé que vous spécifiez figure dans un compte différent de celui que vous utilisez pour configurer un cluster, vous devez spécifier la clé au moyen de son nom ARN.

Le rôle pour le profil d'instance Amazon EC2 doit avoir les autorisations nécessaires pour utiliser la clé CMK que vous spécifiez. Le rôle par défaut pour le profil d'instance dans Amazon EMR est `EMR_EC2_DefaultRole`. Si vous utilisez un autre rôle pour le profil d'instance ou si vous utilisez les rôles IAM pour les demandes EMRFS adressées à Amazon S3, vérifiez que chaque rôle est ajouté en tant qu'utilisateur clé, selon les besoins. Cela donne au rôle les autorisations nécessaires pour utiliser la CMK. Pour de plus amples informations, veuillez consulter Utilisation de stratégies de clé dans le AWS Key Management Service Developer Guide et Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).

Vous pouvez utiliser AWS Management Console pour ajouter votre profil d'instance ou le profil d'instance EC2 à la liste des utilisateurs de clés pour la clé CMK AWS KMS spécifiée, ou vous pouvez utiliser AWS CLI ou un kit SDK AWS pour attacher une stratégie de clé appropriée.

La procédure ci-dessous explique comment ajouter le profil d'instance EMR par défaut, `EMR_EC2_DefaultRole`, en tant qu'utilisateur de clés via AWS Management Console. Elle suppose que vous avez déjà créé une clé CMK. Pour créer une nouvelle clé CMK, consultez Création de clés dans le AWS Key Management Service Developer Guide.

Pour ajouter le profil d'instance EC2 pour Amazon EMR à la liste des utilisateurs de clés de chiffrement

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Sélectionnez l'alias de la clé CMK à modifier.
4. Sur la page de détails de la clé, sous Key Users (Utilisateurs de clés), choisissez Add (Ajouter).
5. Dans la boîte de dialogue Ajouter des utilisateurs clés sélectionnez le rôle approprié. Le nom du rôle par défaut est `EMR_EC2_DefaultRole`.
6. Choisissez Ajouter.

Activation du chiffrement EBS en fournissant des autorisations supplémentaires pour les clés CMK AWS KMS

À partir de Amazon EMR version 5.24.0, vous pouvez chiffrer le périphérique racine EBS et les volumes de stockage à l'aide d'une option de configuration de sécurité. Pour activer cette option, vous devez spécifier AWS KMS comme fournisseur de clés. En outre, vous devez accorder au rôle de service EMR `EMR_DefaultRole` des autorisations pour utiliser la clé CMK que vous spécifiez.

Vous pouvez utiliser AWS Management Console pour ajouter le rôle de service EMR à la liste des utilisateurs de clés pour la clé CMK AWS KMS spécifiée ; vous pouvez utiliser l'AWS CLI ou un kit SDK AWS pour attacher une stratégie de clé appropriée.

La procédure ci-dessous explique comment ajouter le rôle de service EMR par défaut, `EMR_DefaultRole`, en tant qu'utilisateur de clés via AWS Management Console. Elle suppose que vous avez déjà créé une clé CMK. Pour créer une nouvelle clé CMK, consultez Création de clés dans le AWS Key Management Service Developer Guide.

Pour ajouter le rôle de service EMR à la liste des utilisateurs de clé de chiffrement

1. Connectez-vous à la AWS Management Console et ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
2. Pour changer de région AWS, utilisez le sélecteur de région dans l'angle supérieur droit de la page.
3. Sélectionnez l'alias de la clé CMK à modifier.
4. Sur la page de détails de la clé, sous Key Users (Utilisateurs de clés), choisissez Add (Ajouter).
5. Dans la boîte de dialogue Ajouter des utilisateurs clés sélectionnez le rôle approprié. Le nom du rôle de service EMR par défaut est `EMR_DefaultRole`.
6. Choisissez Add (Ajouter).

Création d'un fournisseur de clés personnalisé

Lorsque vous utilisez une configuration de sécurité, vous devez spécifier un autre nom de classe de fournisseur pour le chiffrement de disque local et le chiffrement Amazon S3.

Lorsque vous créez un fournisseur de clés personnalisé, l'application doit mettre en œuvre l'interface EncryptionMaterialsProvider, qui est disponible dans le kit AWS SDK for Java version 1.11.0 ou ultérieure. Cette mise en œuvre peut utiliser n'importe quelle stratégie pour fournir les supports de chiffrement. Vous pouvez, par exemple, choisir de fournir des supports de chiffrement statique ou d'intégrer un système de gestion de clés plus complexe.

L'algorithme de chiffrement utilisé pour les supports de chiffrement personnalisés doit être AES/GCM/NoPadding.

La classe EncryptionMaterialsProvider récupère les supports de chiffrement par contexte de chiffrement. Amazon EMR renseigne les informations de contexte de chiffrement au moment de l'exécution pour aider le mandataire à déterminer les supports de chiffrement corrects à renvoyer.

Example Exemple : Utilisation d'un fournisseur de clés personnalisé pour le chiffrement Amazon S3 avec EMRFS

Lorsqu'Amazon EMR extrait les supports de chiffrement de la classe EncryptionMaterialsProvider pour effectuer le chiffrement, EMRFS renseigne éventuellement l'argument materialsDescription avec deux champs : l'URI Amazon S3 de l'objet et l'ID JobFlowId du cluster, qui peuvent être utilisés par la classe EncryptionMaterialsProvider pour renvoyer les supports de chiffrement de façon sélective.

Par exemple, le fournisseur peut renvoyer des clés différentes pour différents préfixes d'URI Amazon S3. La description des supports de chiffrement renvoyés est finalement stockée avec l'objet Amazon S3 plutôt que la valeur materialsDescription qui est générée par EMRFS et transmise au fournisseur. Lors

du déchiffrement d'un objet Amazon S3, la description des supports de chiffrement est transmise à la classe EncryptionMaterialsProvider, afin qu'elle puisse, à nouveau, renvoyer de façon sélective la clé correspondante pour déchiffrer l'objet.

Une mise en œuvre de référence de la classe EncryptionMaterialsProvider est fournie ci-dessous. Un autre fournisseur personnalisé, EMRFSRSAEncryptionMaterialsProvider, est disponible à partir de GitHub.

```
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.EncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.KMSEncryptionMaterials;
import org.apache.hadoop.conf.Configurable;
import org.apache.hadoop.conf.Configuration;

import java.util.Map;

/**
 * Provides KMSEncryptionMaterials according to Configuration
 */
public class MyEncryptionMaterialsProviders implements EncryptionMaterialsProvider,
 Configurable{
  private Configuration conf;
  private String kmsKeyId;
  private EncryptionMaterials encryptionMaterials;

  private void init() {
    this.kmsKeyId = conf.get("my.kms.key.id");
    this.encryptionMaterials = new KMSEncryptionMaterials(kmsKeyId);
  }

  @Override
  public void setConf(Configuration conf) {
    this.conf = conf;
    init();
  }

  @Override
  public Configuration getConf() {
    return this.conf;
  }

  @Override
  public void refresh() {

  }

  @Override
  public EncryptionMaterials getEncryptionMaterials(Map<String, String>
 materialsDescription) {
    return this.encryptionMaterials;
  }

  @Override
  public EncryptionMaterials getEncryptionMaterials() {
    return this.encryptionMaterials;
  }
}
```

Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR

Avec la version 4.80.0 de Amazon EMR ou une version ultérieure, deux options s'offrent à vous afin de spécifier les artefacts pour le chiffrement des données en transit à l'aide d'une configuration de sécurité :

- Vous pouvez créer manuellement les certificats PEM, les inclure dans un fichier zip, puis référencer le fichier zip dans Amazon S3.
- Vous pouvez implémenter un fournisseur de certificats personnalisés en tant que classe Java. Vous spécifiez le fichier JAR de l'application dans Amazon S3 puis vous fournissez le nom de classe complet du fournisseur, comme déclaré dans l'application. Cette classe doit implémenter l'interface TLSArtifactsProvider disponible à partir de la version 1.11.0 de AWS SDK for Java.

Amazon EMR télécharge automatiquement les objets sur chaque nœud du cluster et les utilise ultérieurement pour mettre en œuvre les fonctionnalités de chiffrement open source en transit. Pour plus d'informations sur les options disponibles, consultez Chiffrement en transit (p. 202).

Utilisation des certificats PEM

Lorsque vous spécifiez un fichier zip pour le chiffrement en transit, la configuration de sécurité s'attend à ce que les fichiers PEM contenus dans ce fichier zip aient exactement le même nom que ci-dessous :

Certificats de chiffrement en transit

Nom de fichier	Obligatoire/facultatif	Details (Détails)
privateKey.pem	: obligatoire.	Clé privée
certificateChain.pem	: obligatoire.	Chaîne de certificats
trustedCertificates.pem	- Facultatif.	Obligatoire si le certificat fourni n'est signé ni par l'autorité de certification racine Java par défaut de confiance (CA), ni par un intermédiaire CA qui est rattaché avec l'autorité de certification racine Java par défaut. Les autorités de certification racine Java par défaut sont disponibles : `jre/lib/security/cacerts`.

Nous vous recommandons de configurer le fichier PEM de la clé privée de sorte qu'il joue le rôle de certificat générique permettant d'accéder au domaine Amazon VPC dans lequel se trouvent vos instances de cluster. Par exemple, si le cluster se trouve dans us-east-1 (N. Virginia), vous pourriez définir un nom commun dans la configuration du certificat qui autorise l'accès au cluster en spécifiant `CN=*.ec2.internal` dans la définition d'objet de ce certificat. Si votre groupe réside dans us-west-2 (Oregon), vous pouvez préciser `CN=*.us-west-2.compute.internal`. Pour plus d'informations sur la configuration du cluster EMR dans Amazon VPC, voir Sélectionner un Amazon VPC Sous-réseau du cluster.

L'exemple suivant montre comment utiliser OpenSSL pour générer un certificat X.509 avec une clé privée RSA 1024 bits. La clé autorise l'accès aux instances de cluster Amazon EMR de l'émetteur dans la région us-west-2 (Oregon), comme spécifié par le nom de domaine *`*.us-west-2.compute.internal`* comme nom commun.

D'autres éléments d'objet facultatifs tels que le pays (C), l'état (S), la langue (L), etc. sont spécifiés. Comme un certificat auto-signé est généré, la deuxième commande dans l'exemple copie le fichier `certificateChain.pem` dans le fichier `trustedCertificates.pem`. La troisième commande utilise `zip` pour créer le fichier `my-certs.zip` qui contient les certificats.

> **Important**
>
> Cet exemple est la démonstration d'une preuve de concept uniquement. L'utilisation de certificats auto-signés n'est pas recommandé et présente un risque de sécurité potentiel. Pour les systèmes de production, utilisez une autorité de certification (CA) approuvée pour émettre des certificats.

```
$ openssl req -x509 -newkey rsa:1024 -keyout privateKey.pem -out certificateChain.pem
 -days 365 -nodes -subj '/C=US/ST=Washington/L=Seattle/O=MyOrg/OU=MyDept/CN=*.us-
west-2.compute.internal'
$ cp certificateChain.pem trustedCertificates.pem
$ zip -r -X my-certs.zip certificateChain.pem privateKey.pem trustedCertificates.pem
```

AWS Identity and Access Management pour Amazon EMR

AWS Identity and Access Management (IAM) est un service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux ressources AWS. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (disposant des autorisations) à utiliser les ressources Amazon EMR. IAM est un service AWS que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Audience (p. 207)
- Authentification avec les identités (p. 208)
- Gestion de l'accès à l'aide des stratégies (p. 209)
- Comment Amazon EMR fonctionne avec IAM (p. 211)
- Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213)
- Exemples de stratégies basées sur l'identité Amazon EMR (p. 240)

Audience

Votre utilisation d'AWS Identity and Access Management (IAM) évolue selon la tâche que vous réalisez dans Amazon EMR.

Utilisateur du service – Si vous utilisez le service Amazon EMR pour effectuer votre tâche, votre administrateur fournit les informations d'identification et les autorisations dont vous avez besoin. Plus vous utiliserez de fonctionnalités Amazon EMR pour effectuer votre tâche, plus vous pourrez avoir besoin d'autorisations supplémentaires. Comprendre la gestion des accès peut vous aider à demander à votre administrateur les autorisations appropriées.

Administrateur du service –Si vous êtes le responsable des ressources Amazon EMR de votre entreprise, vous bénéficiez probablement d'un accès complet à Amazon EMR. C'est à vous de déterminer les fonctionnalités et les ressources Amazon EMR auxquelles vos employés pourront accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec Amazon EMR, consultez Comment Amazon EMR fonctionne avec IAM (p. 211).

Administrateur IAM – Si vous êtes un administrateur IAM, vous voudrez peut-être obtenir des détails sur la façon dont vous pouvez écrire des stratégies pour gérer l'accès à Amazon EMR. Pour obtenir des exemples de stratégies Amazon EMR basées sur l'identité que vous pouvez utiliser dans IAM, consultez Exemples de stratégies basées sur l'identité Amazon EMR (p. 240).

Authentification avec les identités

L'authentification correspond au processus par lequel vous vous connectez à AWS via vos informations d'identification. Pour plus d'informations sur la signature à l'aide d'AWS Management Console, consultez La console et la page de connexion IAM dans le IAM Guide de l'utilisateur.

Vous devez être authentifié (connecté à AWS) en tant que Utilisateur racine d'un compte AWS ou utilisateurIAM, ou en assumant un rôle IAM. Vous pouvez également utiliser l'authentification de connexion unique de votre entreprise ou vous connecter via Google ou Facebook. Dans ce cas, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS avec des informations d'identification d'une autre entreprise, vous assumez indirectement un rôle.

Pour vous connecter directement à la AWS Management Console, utilisez votre mot de passe avec votre e-mail utilisateur racine ou votre nom d'utilisateur IAM. Vous pouvez accéder à AWS par programmation avec vos clés d'accès utilisateur utilisateur racine ou IAM. AWS fournit un kit de développement logiciel (SDK) et des outils de ligne de commande pour signer de manière cryptographique votre requête avec vos informations d'identification. Si vous n'utilisez pas les outils AWS, vous devez signer la demande vous-même. Pour ce faire, utilisez Signature Version 4, un protocole permettant d'authentifier les demandes d'API entrantes. Pour plus d'informations sur l'authentification des demandes, consultez Processus de signature Signature Version 4 dans le AWS General Reference.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être également fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multi-facteurs (MFA) pour améliorer la sécurité de votre compte. Pour en savoir plus, consultez Utilisation de Multi-Factor Authentication (MFA) dans AWS dans le IAM Guide de l'utilisateur.

Utilisateur racine d'un compte AWS

Lorsque vous créez un compte AWS, vous commencez avec une seule identité de connexion disposant d'un accès complet à tous les services et ressources AWS du compte. Cette identité est appelée la utilisateur racinedu compte AWS et elle est accessible après connexion à l'aide de l'adresse e-mail et du mot de passe utilisés pour la création du compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes, y compris pour les tâches administratives. Au lieu de cela, respectez la bonne pratique qui consiste à avoir recours à l'utilisateur racine uniquement pour créer le premier utilisateurIAM. Ensuite, mettez en sécurité les informations d'identification de l'utilisateur racine et utilisez-les pour effectuer uniquement certaines tâches de gestion des comptes et des services.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité dans votre compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou application. Un utilisateur IAM peut disposer d'informations d'identification à long terme comme un nom d'utilisateur et un mot de passe ou un ensemble de clés d'accès. Pour savoir comment générer des clés d'accès, consultez Gestion des clés d'accès pour les utilisateurs IAM dans le IAM Guide de l'utilisateur. Lorsque vous générez des clés d'accès pour un utilisateur IAM, veillez à afficher et enregistrer la paire de clés de manière sécurisée. Vous ne pourrez plus récupérer la clé d'accès secrète à l'avenir. Au lieu de cela, vous devrez générer une nouvelle paire de clés d'accès.

Un groupe IAM est une identité qui spécifie un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé Admins IAM et accorder à ce groupe les autorisations leur permettant d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus de détails, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le IAM Guide de l'utilisateur.

Rôles IAM

Un rôle IAM est une entité au sein de votre compte AWS qui dispose d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais un rôle n'est pas associé à une personne en particulier. Vous pouvez temporairement endosser un rôle IAM dans l'AWS Management Console grâce au changement de rôle. Vous pouvez obtenir un rôle en appelant une opération d'API AWS CLI ou AWS à l'aide d'une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le IAM Guide de l'utilisateur.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Autorisations utilisateur IAM temporaires – Un utilisateur IAM peut endosser un rôle IAM pour accepter différentes autorisations temporaires concernant une tâche spécifique.
- Accès d'utilisateurs fédérés – Au lieu de créer un utilisateur IAM, vous pouvez utiliser des identités d'utilisateur préexistantes provenant d'AWS Directory Service, de l'annuaire d'utilisateurs de votre entreprise ou d'un fournisseur d'identité web. On parle alors d'utilisateurs fédérés. AWS attribue un rôle à un utilisateur fédéré lorsque l'accès est demandé via un fournisseur d'identité. Pour plus d'informations sur les utilisateurs fédérés, consultez Utilisateurs fédérés et rôles dans le IAM Guide de l'utilisateur.
- Accès entre comptes – Vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (mandataire de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès entre plusieurs comptes. Toutefois, certains services AWS vous permettent d'attacher une stratégie directement à une ressource (au lieu d'utiliser un rôle en tant que proxy). Pour en savoir plus sur la différence entre les rôles et les stratégies basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les stratégies basées sur les ressources dans le IAM Guide de l'utilisateur.
- Accès à un service AWS –Un rôle de service est un rôle IAM qu'un service assume pour effectuer des actions dans votre compte en votre nom. Lorsque vous configurez certains environnements de services AWS, vous devez définir un rôle que ce service devra assumer. Ce rôle de service doit comprendre toutes les autorisations nécessaires pour que le service puisse accéder aux ressources AWS dont il a besoin. Les rôles de service varient d'un service à un service, mais nombre d'entre eux vous permettent de choisir vos autorisations, tant que vous respectez les exigences documentées pour le service en question. Les rôles de service fournissent un accès uniquement au sein de votre compte et ne peuvent pas être utilisés pour accorder l'accès à des services dans d'autres comptes. Vous créez, modifiez et supprimez un rôle de service à partir d'IAM. Par exemple, vous pouvez créer un rôle qui permet à Amazon Redshift d'accéder à un compartiment Amazon S3 en votre nom, puis de charger les données stockées dans ce compartiment dans un cluster Amazon Redshift. Pour plus d'informations, consultez Création d'un rôle pour déléguer des autorisations à un service AWS dans le IAM Guide de l'utilisateur.
- Applications qui s'exécutent sur Amazon EC2 –Vous pouvez utiliser un rôle IAM pour gérer des informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et effectuent des demandes d'API AWS CLI ou AWS. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un rôle AWS à une instance EC2 et le rendre disponible à toutes les applications associées, vous pouvez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le IAM Guide de l'utilisateur.

Pour savoir si vous devez utiliser ces rôles IAM ou non, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le IAM Guide de l'utilisateur.

Gestion de l'accès à l'aide des stratégies

Vous contrôler les accès dans AWS en créant des stratégies et en les attachant à des identités IAM ou à des ressources AWS. Une stratégie est un objet dans AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit les autorisations de ces dernières. AWS évalue ces stratégies lorsqu'une entité

(utilisateur racine, utilisateur IAM ou rôle IAM) envoie une demande. Les autorisations dans les stratégies déterminent si la demande est autorisée ou refusée. La plupart des stratégies sont stockées dans AWS en tant que documents JSON. Pour plus d'informations sur la structure et le contenu du document de stratégie JSON, consultez Présentation des stratégies JSON dans le IAM Guide de l'utilisateur.

Les stratégies permettent à un administrateur IAM de spécifier qui a accès aux ressources AWSet quelles actions ces personnes peuvent exécuter sur ces ressources. Chaque entité IAM (utilisateur ou rôle) démarre sans autorisation. En d'autres termes, par défaut, les utilisateurs ne peuvent rien faire, pas même changer leurs propres mots de passe. Pour autoriser un utilisateur à effectuer une opération, un administrateur doit associer une stratégie d'autorisations à ce dernier. Il peut également ajouter l'utilisateur à un groupe disposant des autorisations prévues. Lorsqu'un administrateur accorde des autorisations à un groupe, tous les utilisateurs de ce groupe se voient octroyer ces autorisations.

Les stratégies IAM définissent les autorisations d'une action quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une stratégie qui autorise l'action `iam:GetRole`. Un utilisateur avec cette stratégie peut obtenir des informations utilisateur à partir de l'AWS Management Console, de l'AWS CLI ou de l'API AWS.

Stratégies basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un rôle ou un groupe IAM. Ces stratégies contrôlent les actions que peut exécuter cette identité, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une stratégie basée sur l'identité, consultez Création de stratégies IAM dans le IAM Guide de l'utilisateur.

Les stratégies basées sur l'identité peuvent être classées comme étant des stratégies en ligne ou des stratégies gérées. Les stratégies en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les stratégies gérées sont des stratégies autonomes que vous pouvez lier à plusieurs utilisateurs, groupes et rôles de votre compte AWS. Les stratégies gérées incluent les stratégies gérées par AWS et les stratégies gérées par le client. Pour découvrir comment choisir entre une stratégie gérée ou une stratégie en ligne, consultez Choix entre les stratégies gérées et les stratégies en ligne dans le IAM Guide de l'utilisateur.

Autres types de stratégie

AWS prend en charge d'autres types de stratégies moins courantes. Ces types de stratégies peuvent définir le nombre maximal d'autorisations qui vous sont accordées par des types de stratégies plus courants.

- Limite d'autorisations – Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez les autorisations maximales qu'une stratégie basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations obtenues représentent la combinaison des stratégies basées sur l'identité de l'entité et de ses limites d'autorisations. Les stratégies basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le IAM Guide de l'utilisateur.
- Stratégies de contrôle de service (SCP) – Les SCP sont des stratégies JSON qui spécifient le nombre maximal d'autorisations pour une organisation ou une unité d'organisation (OU) dans AWS Organizations. AWS Organizations est un service qui vous permet de regrouper et de gérer de façon centralisée plusieurs comptes AWS détenus par votre entreprise. Si vous activez toutes les fonctions d'une organisation, vous pouvez appliquer les stratégies de contrôle de service (SCP) à l'un ou à l'ensemble de vos comptes. La SCP limite les autorisations pour les entités dans les comptes membres, y compris dans chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organisations et les SCP, consultez Fonctionnement des stratégies de contrôle de service dans le Manuel de l'utilisateur AWS Organizations.
- Stratégies de session – Les stratégies de session sont des stratégies avancées que vous transmettez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle

ou un utilisateur fédéré. Les autorisations de la session obtenue sont une combinaison des stratégies basées sur l'identité de l'utilisateur ou du rôle et des stratégies de session. Les autorisations peuvent également provenir d'une stratégie basée sur les ressources. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour de plus amples informations, veuillez consulter Stratégies de session dans le IAM Guide de l'utilisateur.

Plusieurs types de stratégie

Lorsque plusieurs types de stratégies s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour découvrir la façon dont AWS détermine s'il convient d'autoriser une demande en présence de plusieurs types de stratégies, consultez Logique d'évaluation de stratégies dans le IAM Guide de l'utilisateur.

Comment Amazon EMR fonctionne avec IAM

Avec les stratégies basées sur une identité IAM, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Amazon EMR prend en charge des actions, ressources et clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une stratégie JSON, consultez Références des éléments de stratégie JSON IAM dans le IAM Guide de l'utilisateur.

Amazon EMR ne prend pas en charge les stratégies basées sur les ressource.

Actions

L'élément `Action` d'une stratégie basée sur une identité IAM décrit les actions spécifiques qui seront autorisées ou refusées par la stratégie. Les actions de stratégie possèdent généralement le même nom que l'opération d'API AWS associée. L'action est utilisée dans une stratégie pour permettre d'effectuer l'opération associée.

Actions de politique dans Amazon EMR utilisez le préfixe suivant avant l'action : `elasticmapreduce:`. Par exemple, pour accorder à quelqu'un l'autorisation de créer un cluster à l'aide de l' `RunJobFlow` de l'opération d'API, vous incluez le `elasticmapreduce:RunJobFlow` dans leur stratégie. Les déclarations de stratégie doivent inclure un élément `Action` ou `NotAction`. Amazon EMR définit son propre ensemble d'actions qui décrivent les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit.

```
"Action": [
      "elasticmapreduce:action1",
      "elasticmapreduce:action2"
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante.

```
"Action": "elasticmapreduce:Describe*"
```

Pour afficher la liste des actions Amazon EMR, veuillez consulter Actions Defined by Amazon EMR dans le IAM Guide de l'utilisateur.

Resources

L'élément `Resource` spécifie les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Vous spécifiez une ressource à l'aide d'un ARN ou du caractère générique (*) pour indiquer que l'instruction s'applique à toutes les ressources.

Pour voir la liste des types de ressources Amazon EMR et de leurs ARN, veuillez consulter Resources Defined by Amazon EMR dans le IAM Guide de l'utilisateur. Pour en savoir plus sur les actions avec laquelle vous pouvez spécifier l'ARN de chaque ressource, consultez Actions Defined by Amazon EMR.

Clés de condition

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, comme égal ou inférieur, pour faire correspondre la condition de la stratégie aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une opération `OR` logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM . Pour plus d'informations, consultez IAM Éléments de la politique : Variables et balises dans le Guide de l'utilisateur IAM.

Amazon EMR définit son propre ensemble de clés de condition et prend également en charge l'utilisation des clés de condition générales. Pour voir toutes les clés de condition globales AWS, consultez Clés de contexte de condition globales AWS dans le IAM Guide de l'utilisateur.

Toutes les actions Amazon EC2 prennent en charge les clés de condition `aws:RequestedRegion` et `ec2:Region`. Pour plus d'informations, consultez Exemple : Restriction de l'accès à une région spécifique.

Pour voir une liste de clés de conditions Amazon EMR, consultez Condition Keys for Amazon EMR dans le IAM Guide de l'utilisateur. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions Defined by Amazon EMR.

Utiliser les balises de cluster et de bloc-notes avec des stratégies de contrôle d'accès IAM

L'autorisation d'actions Amazon EMR associées à Blocs-notes EMR et à des clusters EMR peut être ajustée à l'aide d'un contrôle d'accès basé sur des balises avec les stratégies IAM basées sur les identités. Vous pouvez utiliser des clés de condition dans un élément `Condition` (également nommé bloc `Condition`) pour autoriser certaines actions uniquement lorsqu'un bloc-notes, un cluster ou les deux ont une clé de balise ou une combinaison clé-valeur spécifique. Vous pouvez également limiter l'action `CreateEditor` (qui crée un Bloc-notes EMR) et l'action `RunJobFlow` (qui crée un cluster) afin qu'une demande de balise soit soumise lors de la création de la ressource.

Dans Amazon EMR, les clés de condition qui peuvent être utilisées dans un élément `Condition` s'appliquent uniquement aux actions d'API Amazon EMR où `ClusterID` ou `NotebookID` est un paramètre de demande obligatoire. Par exemple, l'action ModifyInstanceGroups ne prend pas en charge les clés de contexte, car `ClusterID` est un paramètre facultatif.

Lorsque vous créez un bloc-notes EMR, une balise par défaut est appliquée avec une chaîne de clé de `creatorUserId` définie sur la valeur de l'ID utilisateur IAM qui a créé le bloc-notes. Cela est utile pour limiter les actions autorisées pour le bloc-notes uniquement au créateur.

Les clés de condition suivantes sont disponibles dans Amazon EMR:

- Utilisez la clé de contexte de condition `elasticmapreduce:ResourceTag/`*`TagKeyString`* pour autoriser ou refuser des actions d'utilisateur sur des clusters ou des bloc-notes avec des balises dotées de la *`TagKeyString`* que vous spécifiez. Si une action transmet à la fois `ClusterID` et `NotebookID`, la condition s'applique à la fois au cluster et au bloc-notes. Cela signifie que les deux ressources doivent

disposer de la chaîne de clé de valeur ou d'une combinaison clé-valeur que vous spécifiez. Vous pouvez utiliser l'élément `Resource` pour limiter l'instruction afin qu'elle s'applique uniquement aux clusters ou blocs-notes, selon les besoins. Pour plus d'informations, consultez Exemples de stratégies basées sur l'identité Amazon EMR (p. 240),

- Utilisez la clé de contexte de condition `elasticmapreduce:RequestTag/TagKeyString` pour exiger une balise spécifique avec des actions/appels d'API. Par exemple, vous pouvez utilisez cette clé de contexte de condition avec l'action `CreateEditor` pour exiger qu'une clé avec `TagKeyString` soit appliquée à un bloc-notes lorsqu'il est créé.

Présentation du paramètre EMR Cluster VisibleToAllUsers

`VisibleToAllUsers` est un paramètre de cluster unique à Amazon EMR qui fonctionne en conjonction avec IAM des stratégies. La valeur par défaut, `true`indique que tous les utilisateurs IAM du compte AWS peuvent effectuer des actions de cluster s'ils disposent des autorisations de stratégie IAM appropriées. Si défini sur `false`, seul l'utilisateur IAM qui a créé le cluster peut effectuer des actions. Vous pouvez remplacer le paramètre par défaut de `true` lorsque vous créez un cluster. Avec le Amazon EMR , vous pouvez le faire en effaçant la Cluster visible pour tous les utilisateurs IAM du compte sur le Sécurité lorsque vous utilisez les options avancées de création de cluster. Avec le AWS CLI, vous pouvez utiliser le `--no-visible-to-all-users` de l'option à créer. Avec l'API EMR, vous pouvez utiliser le Visible à tous les utilisateurs propriété du `RunJobFlow` action. Sur un cluster en cours d'exécution, seul l'utilisateur IAM qui a créé le cluster ou l'utilisateur racine du compte AWS peut modifier le paramètre et l' AWS CLI ou API EMR doit être utilisé. Avec le AWS CLI, vous pouvez utiliser le `--visible-to-all-users` et `--no-visible-to-all-users` des options du avec emr modifier-attributs-cluster. Avec l'API EMR, vous pouvez utiliser le DéfinirVisibleÀTousUtilisateurs action.

Examples

Pour voir des exemples de stratégies Amazon EMR basées sur l'identité, veuillez consulter Exemples de stratégies basées sur l'identité Amazon EMR (p. 240).

Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources

Amazon EMR et les applications telles que Hadoop et Spark doivent obtenir des autorisation d'accéder aux autres ressources AWS et d'exécuter des actions lors de l'exécution. Chaque cluster dans Amazon EMR doit avoir un service role et un rôle pour le Amazon EC2 instance profile. Le IAM les politiques jointes à ces rôles permettent au cluster d'interagir avec les autres AWS services au nom d'un utilisateur.

Un rôle supplémentaire, le rôle Auto Scaling, est obligatoire si votre cluster utilise le dimensionnement automatique dans Amazon EMR. Le rôle de service AWS pour Blocs-notes EMR est obligatoire si vous utilisez Blocs-notes EMR. Pour plus d'informations, consultez Rôles IAM et Utilisation des profils d'instance dans le IAM Guide de l'utilisateur.

Amazon EMR fournit des rôles par défaut et des stratégies gérées par défaut pour chaque rôle qui déterminent les autorisations. Les stratégies gérées sont créées et gérées par AWS, afin qu'elles soient mises à jour automatiquement si les exigences de service évoluent.

Si vous créez un cluster ou un bloc-notes pour la première fois dans un compte, les rôles pour Amazon EMR n'existent pas encore. Une fois que vous les avez créés, vous pouvez voir les rôles, les stratégies correspondantes et les autorisations accordées ou refusées par les stratégies dans la console IAM (https://console.aws.amazon.com/iam/). Vous pouvez spécifier des rôles par défaut à créer et utiliser pour Amazon EMR, vous pouvez créer vos propres rôles et les spécifier individuellement lorsque vous créez un cluster pour personnaliser les autorisations, et vous pouvez spécifier des rôles par défaut à utiliser lors de la création d'un cluster à l'aide de l'AWS CLI. Pour plus d'informations, consultez Customize IAM Roles (p. 231),

Modifying Identity-Based Policies for Permissions to Pass Service Roles for Amazon EMR

Les utilisateurs du cluster ont besoin d'autorisations pour que les applications puissent transmettre les rôles de service pour Amazon EMR en leur nom. La stratégie d'autorisations `AmazonElasticMapReduceFullAccess`, qui est la stratégie gérée par défaut permettant aux utilisateurs d'avoir des autorisations complètes pour Amazon EMR, inclut une instruction qui accorde les autorisations `iam:PassRole` pour toutes les ressources. Cette instruction autorise l'utilisateur à transmettre un rôle quelconque aux autres services AWS afin qu'Amazon EMR puisse interagir avec ces services au nom de l'utilisateur.

Pour mettre en œuvre une stratégie plus restrictive, attachez une stratégie en ligne aux utilisateurs ou groupes appropriés, qui accorde l'autorisation `iam:PassRole` uniquement pour les rôles spécifiques à Amazon EMR. L'exemple suivant illustre une instruction qui accorde l'autorisation `iam:PassRole` uniquement pour les rôles Amazon EMR par défaut : `EMR_DefaultRole`, `EMR_EC2_DefaultRole` et `EMR_AutoScalingDefaultRole`. Si vous utilisez des rôles personnalisés, remplacez les noms de rôles par défaut par vos noms de rôles personnalisés.

```
{
    "Action": "iam:PassRole",
    "Effect": "Allow",
    "Resource": [
        "arn:aws:iam::*:role/EMR_DefaultRole",
        "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
        "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
        "arn:aws:iam::*:role/EMR_Notebooks_DefaultRole"
    ]
}
```

Service Role Summary

Le tableau suivant répertorie les rôles de service IAM associés à Amazon EMR pour une référence rapide.

Function	Rôle par défaut	Description :	Stratégie gérée par défaut
Service Role for Amazon EMR (EMR Role) (p. 218)	`EMR_DefaultRole`	Permet à Amazon EMR d'appeler d'autres services AWS en votre nom lors de la mise en service des ressources et de l'exécution des actions de niveau de service. Ce rôle est obligatoire pour tous les clusters.	`AmazonElasticMapReduceRole` **Important** La demande d'instances Spot requiert un rôle lié à un service. Si ce rôle n'existe pas, le rôle de service EMR doit avoir l'autorisation de le créer ou une erreur d'autorisation se produit. La stratégie gérée comprend une instruction pour

Function	Rôle par défaut	Description :	Stratégie gérée par défaut
			autoriser cette action. Si vous personnalisez ce rôle ou cette stratégie, veillez à inclure une instruction qui permet la création de ce rôle lié à un service. Pour plus d'informations, consultez Service Role for Amazon EMR (EMR Role) (p. 218) et Rôle lié à un service pour les demandes d'instances Spot dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219)	`EMR_EC2_DefaultRole`	Les processus d'application qui s'exécutent au-dessus de l'écosystème Hadoop sur les instances de cluster utilisent ce rôle lorsqu'ils appellent d'autres services AWS. Pour accéder aux données dans Amazon S3 à l'aide d'EMRFS, vous pouvez spécifier différents rôles à endosser en fonction de l'utilisateur ou du groupe qui envoie la requête, ou de l'emplacement des données dans Amazon S3. Pour plus d'informations, consultez Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234), Ce rôle est obligatoire pour tous les clusters.	`AmazonElasticMapReduceforEC2R` Pour plus d'informations, voir Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).

Function	Rôle par défaut	Description :	Stratégie gérée par défaut
Service Role for Automatic Scaling in EMR (Auto Scaling Role) (p. 224)	`EMR_AutoScaling_DefaultRole`	Permet les actions supplémentaires pour les environnements à dimensionnement dynamique. Obligatoire uniquement pour les clusters qui utilisent le dimensionnement automatique dans Amazon EMR. Pour plus d'informations, consultez Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401),	`AmazonElasticMapReduceforAuto` Pour plus d'informations, voir Service Role for Automatic Scaling in EMR (Auto Scaling Role) (p. 224).
Service Role for Blocs-notes EMR (p. 225)	`EMR_Notebooks_DefaultRole`	Fournit les autorisations dont un Bloc-notes EMR a besoin pour accéder aux autres ressources AWS et exécuter des actions. Obligatoire uniquement si Blocs-notes EMR est utilisé.	`AmazonElasticMapReduceEditors` Pour plus d'informations, voir Service Role for Blocs-notes EMR (p. 225). `S3FullAccessPolicy` est également attaché par défaut. Le contenu de cette stratégie est présenté ci-dessous. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:*", "Resource": "*" } ] }`

Function	Rôle par défaut	Description :	Stratégie gérée par défaut
Rôle lié à un service (p. 226)	`AWSServiceRoleForEMRCleanup`	Amazon EMR créé automatiquement un rôle lié à un service. Si le service pour Amazon EMR n'a plus la possibilité de nettoyer les ressources Amazon EC2, Amazon EMR peut utiliser ce rôle pour cette action. Si un cluster utilise des instances Spot, la stratégie d'autorisation attachée au Service Role for Amazon EMR (EMR Role) (p. 218) doit autoriser la création d'un rôle lié à un service. Pour plus d'informations, consultez Autorisations des rôles liés à un service pour Amazon EMR (p. 226),	`AmazonEMRCleanupPolicy`

Rubriques

- IAM Service Roles Used By Amazon EMR (p. 217)
- Customize IAM Roles (p. 231)
- Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234)
- Use Resource-Based Policies for Amazon EMR Access to AWS Glue Data Catalog (p. 238)
- Use IAM Roles with Applications That Call AWS Services Directly (p. 239)
- Allow Users and Groups to Create and Modify Roles (p. 240)

IAM Service Roles Used By Amazon EMR

Amazon EMR utilise des rôles de service IAM pour effectuer des actions en votre nom lors de la mise en service des ressources de cluster, l'exécution d'applications, le dimensionnement dynamique de ressources, ainsi que la création et l'exécution d'Blocs-notes EMR. Amazon EMR utilise les rôles suivants lorsqu'il interagit avec d'autres services AWS. Chaque rôle possède une fonction unique dans Amazon EMR. Les rubriques de cette section décrivent la fonction du rôle et fournissent les rôles et la stratégie d'autorisations par défaut pour chaque rôle.

Si vous disposez d'un code d'application sur votre cluster qui appelle des services AWS directement, vous devrez peut-être utiliser le kit SDK pour spécifier les rôles. Pour plus d'informations, consultez Use IAM Roles with Applications That Call AWS Services Directly (p. 239),

Rubriques

- Service Role for Amazon EMR (EMR Role) (p. 218)
- Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219)
- Service Role for Automatic Scaling in EMR (Auto Scaling Role) (p. 224)
- Service Role for Blocs-notes EMR (p. 225)

- Utilisation du rôle lié à un service pour Amazon EMR (p. 226)

Service Role for Amazon EMR (EMR Role)

Le rôle EMR définit les actions autorisées pour Amazon EMR lors de la mise en service de ressources et de l'exécution des tâches de niveau de service qui ne sont pas effectuées dans le cadre de l'exécution d'une instance EC2 dans un cluster. Par exemple, le rôle de service est utilisé pour mettre en service des instances EC2 lorsqu'un cluster est lancé.

- The default role is `EMR_DefaultRole`
- The default managed policy attached to `EMR_DefaultRole` is `AmazonElasticMapReduceRole`

Le contenu de la version 10 d'`AmazonElasticMapReduceRole` est indiqué ci-dessous.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotInstanceRequests",
                "ec2:CreateFleet",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DeleteLaunchTemplate",
                "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:DeleteVolume",
```

```
                "ec2:DescribeVolumeStatus",
                "ec2:DescribeVolumes",
                "ec2:DetachVolume",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListInstanceProfiles",
                "iam:ListRolePolicies",
                "iam:PassRole",
                "s3:CreateBucket",
                "s3:Get*",
                "s3:List*",
                "sdb:BatchPutAttributes",
                "sdb:Select",
                "sqs:CreateQueue",
                "sqs:Delete*",
                "sqs:GetQueue*",
                "sqs:PurgeQueue",
                "sqs:ReceiveMessage",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "application-autoscaling:RegisterScalableTarget",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:Describe*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/spot.amazonaws.com/
AWSServiceRoleForEC2Spot*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "spot.amazonaws.com"
                }
            }
        }
    ]
}
```

Service Role for Cluster EC2 Instances (EC2 Instance Profile)

Le Rôle de service pour les instances EC2 de cluster (également appelé profil d'instance EC2 pour Amazon EMR) est un type de rôle de service unique assigné à chaque instance EC2 dans un cluster Amazon EMR lors du lancement de l'instance. Les processus d'application qui s'exécutent au-dessus de l'écosystème Hadoop assument ce rôle pour les autorisations permettant d'interagir avec d'autres services AWS.

Pour plus d'informations sur les rôles de service pour les instances EC2, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le IAM Guide de l'utilisateur.

> **Important**
>
> Le Rôle de service pour les instances EC2 de cluster par défaut et la stratégie gérée qu'il utilise sont configurés avec des autorisations qui vous permettent de créer un cluster entièrement fonctionnel aussi facilement que possible. Nous vous recommandons vivement de modifier cette stratégie pour fournir les moindres privilèges obligatoires pour votre application. Pour plus d'informations, consultez Creating a Service Role for Cluster EC2 Instances With Least-Privileged Permissions (p. 221),

Default Role and Managed Policy

- The default role is `EMR_EC2_DefaultRole`.
- The default managed policy attached to `EMR_EC2_DefaultRole` is `AmazonElasticMapReduceforEC2Role`

Le contenu de la version 3 d'`AmazonElasticMapReduceforEC2Role` est indiqué ci-dessous.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "cloudwatch:*",
                "dynamodb:*",
                "ec2:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSteps",
                "kinesis:CreateStream",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:MergeShards",
                "kinesis:PutRecord",
                "kinesis:SplitShard",
                "rds:Describe*",
                "s3:*",
                "sdb:*",
                "sns:*",
                "sqs:*",
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ]
        }
```

```
    ]
}
```

Creating a Service Role for Cluster EC2 Instances With Least-Privileged Permissions

En tant que bonne pratique, nous vous recommandons fortement de créer un Rôle de service pour les instances EC2 de cluster et une stratégie d'autorisations de façon à ce qu'ils possèdent les autorisations minimales pour d'autres services AWS dont votre application a besoin.

La stratégie gérée par défaut, `AmazonElasticMapReduceforEC2Role`, fournit des autorisations qui facilitent le lancement d'un cluster initial. Cependant, Amazon EMR ne nécessite pas d'autorisations pour lancer, contrôler et gérer un cluster de base. Si vous lancez un cluster sans autorisations de cette manière, le cluster est toujours créé et les journaux système sont générés et envoyés aux compartiments Amazon S3 détenus par Amazon EMR à l'aide d'une autre méthode d'autorisation. Toutefois, les applications de cluster ne seront pas en mesure d'interagir avec d'autres services AWS. Par exemple, le cluster ne sera pas en mesure de lire ou d'écrire à partir d'Amazon S3.

La déclaration de stratégie ci-dessous fournit des exemples des autorisations requises pour différentes fonctions d'Amazon EMR. Nous vous recommandons d'utiliser ces autorisations afin de créer une stratégie d'autorisations qui restreint l'accès uniquement aux fonctionnalités et aux ressources dont votre cluster a besoin. Tous les énoncés de politique d'exemple utilisent le *us-west-2* la région et la fictive AWS ID de compte *123456789012*Remplacez-les par les bonnes informations pour votre cluster.

Pour plus d'informations sur la création et la spécification des rôles personnalisés, consultez Customize IAM Roles (p. 231).

> **Note**
>
> Si vous créez un rôle EMR personnalité pour EC2, suivez le flux de travail de base, qui crée automatiquement un profil d'instance du même nom. Amazon EC2 vous permet de créer des profils d'instance et des rôles portant des noms différents, mais Amazon EMR ne prend pas en charge cette configuration, qui entraîne une erreur de type « profil d'instance non valide » lorsque vous créez le cluster.

Reading and Writing Data to Amazon S3 Using EMRFS

Lorsqu'une application s'exécutant sur un cluster Amazon EMR référence des données à l'aide du format `s3://mydata`, Amazon EMR utilise EMRFS pour effectuer la demande. Les clusters lisent et écrivent généralement des données sur Amazon S3 de cette manière, et EMRFS utilise les autorisations attachées au Rôle de service pour les instances EC2 de cluster par défaut. Lorsque vous avez plusieurs utilisateurs de cluster et plusieurs magasins de données, vous pouvez accorder aux utilisateurs des autorisations différentes pour les données EMRFS dans Amazon S3. Pour ce faire, vous pouvez utiliser des rôles IAM pour EMRFS. EMRFS peut ainsi assumer plusieurs rôles avec des stratégies d'autorisation différentes en fonction de l'utilisateur ou du groupe qui envoie la demande ou de l'emplacement des données EMRFS dans Amazon S3. Pour plus d'informations, consultez Configure IAM Roles for EMRFS Requests to Amazon S3 (p. 234),

Étant donné que les rôles IAM pour EMRFS basculeront vers les autorisations attachées au Rôle de service pour les instances EC2 de cluster, comme bonne pratique, nous vous recommandons d'utiliser les rôles IAM pour EMRFS, et de limiter le système EMRFS et les autorisations Amazon S3 attachées au' Rôle de service pour les instances EC2 de cluster.

L'exemple de déclaration ci-dessous illustre les autorisations dont EMRFS a besoin pour effectuer des demandes à Amazon S3.

- *my-data-bucket-in-s3-for-emrfs-reads-and-writes* specifies the bucket in Amazon S3 where the cluster reads and writes data and all sub-folders using `/*`. Add only those buckets and folders that your application requires.

- The policy statement that allows `dynamodb` actions is required only if EMRFS consistent view is enabled. *`EmrFSMetadata`* specifies the default folder for EMRFS consistent view. For more information, see Activation de la vue cohérente (p. 85).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:GetBucketVersioning",
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListBucketVersions",
                "s3:ListMultipartUploadParts",
                "s3:PutBucketVersioning",
                "s3:PutObject",
                "s3:PutObjectTagging"
            ],
            "Resource": [
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes",
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:BatchGetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:DescribeTable",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteTable",
                "dynamodb:UpdateTable"
            ],
            "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/EmrFSMetadata"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "dynamodb:ListTables",
                "s3:HeadBucket"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:DeleteMessageBatch",
                "sqs:ReceiveMessage",
```

```
                "sqs:DeleteQueue",
                "sqs:SendMessage",
                "sqs:CreateQueue"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:EMRFS-Inconsistency-*"
        }

}
```

Archiving Log Files to Amazon S3

La déclaration de stratégie suivante autorise le cluster Amazon EMR à archiver les fichiers journaux dans l'emplacement Amazon S3 spécifié. Dans l'exemple ci-dessous, lorsque le cluster a été créé, *s3://MyLoggingBucket/MyEMRClusterLogs* a été spécifié à l'aide du Emplacement du dossier de journal S3 dans la console, en utilisant `--log-uri` option du AWS CLI, ou utiliser le `LogUri` paramètre dans le `RunJobFlow` commande. Pour plus d'informations, consultez Archive Log Files to Amazon S3 (p. 167),

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::MyLoggingBucket/MyEMRClusterLogs/*"
        }
    ]
}
```

Using the Debugging Tools

La déclaration de stratégie suivante autorise les actions qui sont requises si vous activez l'outil de débogage Amazon EMR. L'archivage de fichiers journaux dans Amazon S3, et les autorisations associées indiquées dans l'exemple ci-dessus, sont requis pour le débogage. Pour plus d'informations, consultez Enable the Debugging Tool (p. 170),

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:AWS-ElasticMapReduce-*"
        }
    ]
}
```

Using the AWS Glue Data Catalog

La déclaration de stratégie suivante autorise les actions qui sont nécessaires si vous utilisez l'AWS Glue Data Catalog en tant que métastore pour les applications. Pour plus d'informations, consultez Utilisation de l'AWS Glue Data Catalog en tant que métastore pour Spark SQL, Utilisation de l'AWS Glue Data Catalog en tant que métastore pour Hive, et Utilisation de Presto avec l'AWS Glue Data Catalog dans le Amazon EMR Guide de version.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ]
        }
    ]
}
```

Service Role for Automatic Scaling in EMR (Auto Scaling Role)

Le rôle d'Auto Scaling pour EMR a une fonction similaire au rôle de service, mais permet des actions supplémentaires pour les environnements au dimensionnement dynamique.

- The default role is `EMR_AutoScaling_DefaultRole`.
- The default managed policy attached to `EMR_AutoScaling_DefaultRole` is `AmazonElasticMapReduceforAutoScalingRole`.

Le contenu de la version 1 d'`AmazonElasticMapReduceforAutoScalingRole` est indiqué ci-dessous.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudwatch:DescribeAlarms",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Service Role for Blocs-notes EMR

Chaque Bloc-notes EMR a besoin d'autorisations pour accéder aux autres ressources AWS et exécuter des actions. Les stratégies IAM attachées à ce rôle de service fournissent des autorisations pour que le bloc-notes interagisse avec d'autres services AWS. Lorsque vous créez un carnet à l'aide de AWS Management Console, vous spécifiez un AWS service role. Vous pouvez utiliser le rôle par défaut, `EMR_Notebooks_DefaultRole`, ou spécifiez un rôle que vous créez. Si un bloc-notes n'a pas été créé auparavant, vous pouvez choisir de créer le rôle par défaut.

- The default role is `EMR_Notebooks_DefaultRole`.
- The default managed policy attached to `EMR_Notebooks_DefaultRole` is `AmazonElasticMapReduceEditorsRole`.

Le contenu de la version 1 d'`AmazonElasticMapReduceEditorsRole` est indiqué ci-dessous.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeNetworkInterfaces",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DescribeTags",
                "ec2:DescribeInstances",
                "ec2:DescribeSubnets",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:DescribeCluster"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "aws:elasticmapreduce:editor-id",
                        "aws:elasticmapreduce:job-flow-id"
                    ]
                }
            }
        }
    ]
}
```

Lorsque vous liez des référentiels Git à votre bloc-notes et que vous devez créer un secret pour le référentiel, vous devez ajouter l'autorisation `secretsmanager:GetSecretValue` dans la stratégie IAM attachée au rôle de service pour les blocs-notes EMR. Voici un exemple de stratégie :

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

Utilisation du rôle lié à un service pour Amazon EMR

Amazon EMR utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement à Amazon EMR. Le rôle lié à un service est prédéfini par Amazon EMR et inclut les autorisations qu'Amazon EMR requiert pour appeler Amazon EC2 en votre nom afin de nettoyer les ressources du cluster une fois qu'elles ne sont plus utilisées. Le rôle lié à un service fonctionne avec le rôle du service Amazon EMR et le profil d'instance Amazon EC2 pour Amazon EMR. Pour plus d'informations sur le rôle de service et le profil d'instance, consultez la section Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).

Amazon EMR définit les autorisations de ce rôle lié à un service et, sauf définition contraire, seul Amazon EMR peut endosser ce rôle. Les autorisations définies comprennent la stratégie d'approbation et la stratégie d'autorisations. De plus, cette stratégie d'autorisations ne peut pas être attachée à une autre entité IAM. Vous pouvez supprimer le rôle uniquement après avoir mis fin à tous les clusters EMR du compte.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services qui comportent Oui dans la colonne Rôle lié à un service. Choisissez un Oui ayant un lien permettant de consulter la documentation du rôle lié à un service, pour ce service.

Autorisations des rôles liés à un service pour Amazon EMR

Amazon EMR utilise le rôle AWSServiceRoleForEMRCleanup, qui est un rôle basé sur un service qui permet à Amazon EMR de mettre fin à des ressources Amazon EC2 et de les supprimer en votre nom si le rôle de service Amazon EMR n'est plus en mesure de le faire. Amazon EMR crée automatiquement le rôle lors de la création du cluster s'il n'existe pas déjà.

Le rôle lié à un service AWSServiceRoleForEMRCleanup approuve les services suivants pour endosser le rôle :

- `elasticmapreduce.amazonaws.com`

La stratégie d'autorisations liée au rôle AWSServiceRoleForEMRCleanup permet à Amazon EMR de réaliser les actions suivantes sur les ressources spécifiées :

- Action : `DescribeInstances` sur `ec2`
- Action : `DescribeSpotInstanceRequests` sur `ec2`
- Action : `ModifyInstanceAttribute` sur `ec2`
- Action : `TerminateInstances` sur `ec2`
- Action : `CancelSpotInstanceRequests` sur `ec2`
- Action : `DeleteNetworkInterface` sur `ec2`
- Action : `DescribeInstanceAttribute` sur `ec2`
- Action : `DescribeVolumeStatus` sur `ec2`
- Action : `DescribeVolumes` sur `ec2`

- Action : `DetachVolume` sur `ec2`
- Action : `DeleteVolume` sur `ec2`

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, groupe ou rôle) de créer, modifier ou supprimer un rôle lié à un service.

Pour permettre à une entité IAM de créer le rôle lié à un service AWSServiceRoleForEMRCleanup

Ajoutez l'instruction suivante à la stratégie d'autorisations de l'entité IAM qui doit créer le rôle lié à un service :

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Pour permettre à une entité IAM de modifier la description du rôle lié à un service AWSServiceRoleForEMRCleanup

Ajoutez l'instruction suivante à la stratégie d'autorisations de l'entité IAM qui doit modifier la description d'un rôle lié à un service :

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Pour permettre à une entité IAM de supprimer le rôle lié à un service AWSServiceRoleForEMRCleanup

Ajoutez l'instruction suivante à la stratégie d'autorisations de l'entité IAM qui doit supprimer un rôle lié à un service :

```
{
    "Effect": "Allow",
```

```
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Création d'un rôle lié à un service pour Amazon EMR

Vous n'avez pas besoin de créer manuellement le rôle lié à un service AWSServiceRoleForEMRCleanup. Lorsque vous lancez un cluster, soit pour la première fois, soit lorsqu'un rôle lié à un service n'est pas présent, Amazon EMR crée le rôle lié à un service pour vous. Vous devez disposer des autorisations de création du rôle lié au service. Pour consulter un exemple d'instruction qui ajoute cette fonctionnalité à la stratégie d'autorisations d'une entité IAM (utilisateur, groupe ou rôle, par exemple), consultez la section Autorisations des rôles liés à un service pour Amazon EMR (p. 226).

> **Important**
>
> Si vous utilisiez Amazon EMR avant October 24, 2017, lorsque les rôles liés à un service n'étaient pas pris en charge, Amazon EMR a créé le rôle AWSServiceRoleForEMRCleanup dans votre compte. Pour plus d'informations, consultez Un nouveau rôle est apparu dans mon compte IAM.

Modification d'un rôle lié à un service pour Amazon EMR

Amazon EMR ne vous permet pas de modifier le rôle lié à un service AWSServiceRoleForEMRCleanup. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas modifier le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide de IAM.

Modification de la description d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM, pour modifier la description d'un rôle lié à un service.

Pour modifier la description d'un rôle lié à un service (console)

1. Dans le volet de navigation de la console IAM, choisissez Rôles.
2. Choisissez le nom du rôle à modifier.
3. A l'extrême droite de Role description (Description du rôle), choisissez Edit (Modifier).
4. Saisissez une nouvelle description dans la zone et choisissez Save changes (Enregistrer les modifications).

Modification de la description d'un rôle lié à un service (interface de ligne de commande IAM)

Vous pouvez utiliser les commandes IAM de l'AWS Command Line Interface pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle d'un rôle lié à un service (CLI)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez les commandes suivantes :

```
$ aws iam get-role --role-name role-name
```

Utilisez le nom du rôle, pas l'ARN, pour faire référence aux rôles avec les commandes CLI. Par exemple, si un rôle a l'ARN : `arn:aws:iam::123456789012:role/myrole`, vous faites référence au rôle en tant que **`myrole`**.

2. Pour mettre à jour la description d'un rôle lié à un service, utilisez l'une des commandes suivantes :

```
$ aws iam update-role-description --role-name role-name --description description
```

Modification de la description d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM, pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle d'un rôle lié à un service (API)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez la commande suivante :

   API IAM : GetRole
2. Pour mettre à jour la description d'un rôle, utilisez la commande suivante :

   API IAM : UpdateRoleDescription

Suppression d'un rôle lié à un service pour Amazon EMR

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée non surveillée ou non gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord vérifier qu'aucune session n'est active pour le rôle et supprimer toutes les ressources utilisées par le rôle.

Pour vérifier si une session est active pour le rôle lié à un service dans la console IAM

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Rôles. Sélectionnez le nom (pas la case à cocher) du rôle AWSServiceRoleForEMRCleanup.
3. Sur la page Récapitulatif du rôle sélectionné, choisissez Access Advisor.
4. Dans l'onglet Access Advisor, consultez l'activité récente pour le rôle lié à un service.

   Note

   Si vous n'êtes pas certain que Amazon EMR utilise le rôle AWSServiceRoleForEMRCleanup, vous pouvez essayer de supprimer le rôle. Si le service utilise le rôle, la suppression échoue et vous avez accès aux régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Pour supprimer les ressources Amazon EMR utilisées par le AWSServiceRoleForEMRCleanup

- Mettez fin à tous les clusters de votre compte. Pour plus d'informations, consultez Arrêter un cluster (p. 382).

Suppression d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (console)

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Rôles. Sélectionnez la case à cocher en regard de AWSServiceRoleForEMRCleanup, et non le nom ou la ligne.
3. Pour Actions du rôle en haut de la page, sélectionnez Supprimer le rôle.
4. Dans la boîte de dialogue de confirmation, vérifiez les dernières données consultées dans le service qui indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service AWS. Cela vous permet de confirmer si le rôle est actif actuellement. Pour poursuivre, choisissez Oui, supprimer.
5. Consultez les notifications de la console IAM pour surveiller la progression de la suppression du rôle lié à un service. Dans la mesure où la suppression du rôle lié à un service IAM est asynchrone, une fois que vous soumettez le rôle afin qu'il soit supprimé, la suppression peut réussir ou échouer. Si la tâche échoue, vous pouvez choisir Afficher les détails ou Afficher les ressources à partir des notifications pour connaître le motif de l'échec de la suppression. Si la suppression échoue parce que certaines ressources du service sont actuellement utilisées par le rôle, la raison de l'échec comprend une liste de ressources.

Suppression d'un rôle lié à un service (interface de ligne de commande IAM)

Vous pouvez utiliser les commandes IAM de l'AWS Command Line Interface pour supprimer un rôle lié à un service. Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Si ces conditions ne sont pas satisfaites, cette demande peut être refusée.

Pour supprimer un rôle lié à un service (CLI)

1. Pour vérifier l'état de la tâche de suppression, vous devez capturer l'information `deletion-task-id` dans la réponse. Tapez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service :

   ```
   $ aws iam delete-service-linked-role --role-name AWSServiceRoleForEMRCleanup
   ```

2. Tapez la commande suivante pour vérifier l'état de la tâche de suppression :

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Suppression d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service. Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Si ces conditions ne sont pas satisfaites, cette demande peut être refusée.

Pour supprimer un rôle lié à un service (API)

1. Pour envoyer une demande de suppression d'un rôle lié à un service, appelez DeleteServiceLinkedRole. Dans la demande, spécifiez le nom du rôle AWSServiceRoleForEMRCleanup.

Pour vérifier l'état de la tâche de suppression, vous devez capturer l'information `DeletionTaskId` dans la réponse.

2. Pour vérifier l'état de la suppression, appelez GetServiceLinkedRoleDeletionStatus. Dans la demande, spécifiez l'information `DeletionTaskId`.

   L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Régions prises en charge pour les rôles liés à un service Amazon EMR

Amazon EMR prend en charge l'utilisation des rôles liés à un service vers les régions suivantes.

Nom de la région	Identité de la région	Prise en charge dans Amazon EMR
USA Est (Virginie du Nord)	us-east-1	Oui
USA Est (Ohio)	us-east-2	Oui
USA Ouest (Californie du Nord)	us-west-1	Oui
USA Ouest (Oregon)	us-west-2	Oui
Asie-Pacifique (Mumbai)	ap-south-1	Oui
Asie-Pacifique (Osaka-Local)	ap-northeast-3	Oui
Asie-Pacifique (Séoul)	ap-northeast-2	Oui
Asie-Pacifique (Singapour)	ap-southeast-1	Oui
Asie-Pacifique (Sydney)	ap-southeast-2	Oui
Asie-Pacifique (Tokyo)	ap-northeast-1	Oui
Canada (Centre)	ca-central-1	Oui
Europe (Francfort)	eu-central-1	Oui
Europe (Irlande)	eu-west-1	Oui
Europe (Londres)	eu-west-2	Oui
Europe (Paris)	eu-west-3	Oui
Amérique du Sud (São Paulo)	sa-east-1	Oui

Customize IAM Roles

Vous avez la possibilité de personnaliser les rôles de service IAM et les autorisations pour limiter les privilèges selon vos exigences en matière de sécurité. Pour personnaliser les autorisations, nous vous recommandons de créer de nouveaux rôles et stratégies. Commencez avec les autorisations des stratégies gérées pour les rôles par défaut (par exemple, `AmazonElasticMapReduceforEC2Role` et `AmazonElasticMapReduceRole`). Ensuite, copiez et collez le contenu dans les déclarations de la nouvelle stratégie, modifiez les autorisations selon vos besoins et attachez les stratégies d'autorisations modifiées pour les rôles que vous créez. Vous devez avoir les autorisations IAM appropriées pour travailler

avec les rôles et les stratégies. Pour plus d'informations, consultez Allow Users and Groups to Create and Modify Roles (p. 240),

Si vous créez un rôle EMR personnalité pour EC2, suivez le flux de travail de base, qui crée automatiquement un profil d'instance du même nom. Amazon EC2 vous permet de créer des profils d'instance et des rôles portant des noms différents, mais Amazon EMR ne prend pas en charge cette configuration, qui entraîne une erreur de type « profil d'instance non valide » lorsque vous créez le cluster.

> **Important**
>
> Les stratégies en ligne ne sont pas automatiquement mises à jour lorsque les exigences de service évoluent. Si vous créez et attachez des stratégies en ligne, sachez que des mises à jour de services peuvent se produire soudainement et provoquer des erreurs d'autorisations. Pour de plus amples informations, veuillez consulter Stratégies gérées et stratégies en ligne dans le IAM Guide de l'utilisateur et Specify Custom IAM Roles When You Create a Cluster (p. 232).

Pour plus d'informations sur la manipulation de rôles IAM, consultez les rubriques suivantes dans le IAM Guide de l'utilisateur :

- Creating a Role to Delegate Permissions to an AWS Service
- Modifying a Role
- Deleting a Role

Specify Custom IAM Roles When You Create a Cluster

Vous spécifiez le rôle de service pour Amazon EMR et le rôle pour le profil d'instance Amazon EC2 lorsque vous créez un cluster. L'utilisateur qui crée des clusters a besoin d'autorisations pour récupérer et attribuer des rôles aux instances EC2 et Amazon EMR. Dans le cas contraire, l'erreur Le compte d'utilisateur n'est pas autorisé à appeler EC2 se produit. Pour plus d'informations, consultez Allow Users and Groups to Create and Modify Roles (p. 240),

Use the Console to Specify Custom Roles

Lorsque vous créez un cluster, vous pouvez spécifier un rôle de service personnalisé pour Amazon EMR, un rôle personnalisé pour le profil d'instance EC2 et un rôle Auto Scaling personnalisé à l'aide des Advanced Options (Options avancées). Lorsque vous utilisez les Options rapides, le rôle de service par défaut et le rôle par défaut du profil d'instance EC2 sont spécifiés. Pour plus d'informations, consultez IAM Service Roles Used By Amazon EMR (p. 217),

Pour spécifier des rôles IAM personnalisés à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Choisissez les réglages de cluster appropriés pour votre application jusqu'à ce que vous atteigniez les Options de sécurité.

   Sous Permissions (Autorisations), les rôles Par défaut pour Amazon EMR sont sélectionnés.
4. Choisissez Personnalisé.
5. Pour chaque type de rôle, sélectionnez un rôle dans la liste. Seuls les rôles dans votre compte ayant la stratégie d'approbation appropriée pour ce type de rôle sont répertoriés.

6. Choisissez d'autres options selon les besoins de votre cluster, puis choisissez Créer un cluster.

Use the AWS CLI to Specify Custom Roles

Vous pouvez spécifier un rôle de service EMR et un Rôle de service pour les instances EC2 de cluster explicitement à l'aide d'options avec la commande `create-cluster` depuis l'AWS CLI. Utilisez l'option `--service-role` pour spécifier le rôle de service. Utilisez l'argument `InstanceProfile` de l'option `--ec2-attributes` pour spécifier le rôle pour le profil d'instance EC2.

Le Auto Scaling le rôle est spécifié en utilisant une option distincte, `--auto-scaling-role`. Pour plus d'informations, voir Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401).

Spécification de rôles IAM personnalisés à l'aide de l'AWS CLI.

- La commande suivante spécifie le rôle de service personnalisé, *`MyCustomServiceRoleForEMR`*, et un rôle personnalisé pour le profil d'instance EC2, *`MyCustomServiceRoleForClusterEC2Instances`*, lors du lancement d'un cluster. Cet exemple utilise le rôle Amazon EMR par défaut.

  Note

  Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-5.30.1 \
--applications Name=Hive Name=Pig --service-role MyCustomServiceRoleForEMR \
--ec2-attributes InstanceProfile=MyCustomServiceRoleForClusterEC2Instances,\
KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

Vous pouvez utiliser ces options pour spécifier explicitement les rôles par défaut plutôt qu'à l'aide de l'option `--use-default-roles`. L'option `--use-default-roles` spécifie le rôle de service et le rôle pour le profil d'instance EC2 défini dans le fichier `config` pour l'AWS CLI.

L'exemple suivant montre le contenu d'un fichier `config` pour l'AWS CLI et spécifie les rôles personnalisés pour Amazon EMR. Avec ce fichier de configuration, lorsque `--use-default-roles` est spécifié, le cluster est créé à l'aide du *`MyCustomServiceRoleForEMR`* $and *`MyCustomServiceRoleForClusterEC2Instances`*Par défaut, le fichier `config` spécifie le `service_role` par défaut en tant que `AmazonElasticMapReduceRole` et le `instance_profile` par défaut en tant que `EMR_EC2_DefaultRole`.

```
[default]
output = json
region = us-west-1
aws_access_key_id = myAccessKeyID
aws_secret_access_key = mySecretAccessKey
emr =
     service_role = MyCustomServiceRoleForEMR
     instance_profile = MyCustomServiceRoleForClusterEC2Instances
```

Configure IAM Roles for EMRFS Requests to Amazon S3

Lorsqu'une application s'exécutant sur un cluster Amazon EMR référence des données à l'aide du format `s3://mydata`, Amazon EMR utilise EMRFS pour effectuer la demande. Pour interagir avec Amazon S3, EMRFS suppose que les stratégies d'autorisations attachées au Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219) ont été spécifiées lors de la création du cluster. Le même Rôle de service pour les instances EC2 de cluster est utilisé, quel que soit l'utilisateur ou le groupe, à l'aide de l'application ou de l'emplacement des données dans Amazon S3. Si vous avez des clusters avec plusieurs utilisateurs qui ont besoin de différents niveaux d'accès aux données dans Amazon S3 via EMRFS, vous pouvez définir une configuration de sécurité avec des rôles IAM pour EMRFS. EMRFS peut endosser un autre Rôle de service pour les instances EC2 de cluster en fonction de l'utilisateur ou du groupe qui fait la demande, ou en fonction de l'emplacement des données dans Amazon S3. Chaque rôle IAM pour EMRFS peut avoir différentes autorisations pour accéder aux données dans Amazon S3.

Les rôles IAM pour EMRFS sont disponibles avec Amazon EMR version 5.10.0 et versions ultérieures. Si vous utilisez une version précédente ou si vous avez d'autres exigences en matière d'autorisation au-delà de ce que proposent les rôles IAM pour EMRFS, vous pouvez créer un fournisseur d'informations d'identification personnalisées à la place. Pour plus d'informations, consultez Autorisation d'accès aux données EMRFS dans Amazon S3 (p. 101), Pour plus d'informations sur EMRFS, consultez Utiliser le système de fichiers EMR (EMRFS) (p. 84).

Lorsque vous utilisez une configuration de sécurité pour spécifier des rôles IAM pour EMRFS, vous définissez des mappages de rôle. Chaque mappage de rôle spécifie un rôle IAM qui correspond aux identifiants. Ces identifiants déterminent la base pour accéder à Amazon S3 via EMRFS. Les identifiants peuvent être des utilisateurs, des groupes ou des préfixes Amazon S3 qui indiquent un emplacement de données. Lorsqu'EMRFS effectue une demande à Amazon S3, si la demande correspond à la base d'accès, EMRFS demande aux instances EC2 de cluster d'assumer le rôle IAM correspondant à la requête. Les autorisations IAM attachées à ce rôle s'appliquent au lieu des autorisations IAM attachées au Rôle de service pour les instances EC2 de cluster.

Les utilisateurs et les groupes dans un mappage de rôle sont des utilisateurs et des groupes Hadoop qui sont définis sur le cluster. Les utilisateurs et les groupes sont transmis à EMRFS dans le cadre de l'application à l'aide de celle-ci (par exemple, un emprunt d'identité de l'utilisateur YARN). Le préfixe Amazon S3 peut être un spécificateur de compartiment de quelque profondeur que ce soit (par exemple, `s3://mybucket` ou `s3://mybucket/myproject/mydata`). Vous pouvez spécifier plusieurs identificateurs au sein d'un même mappage de rôle, mais ils doivent tous être du même type.

> **Important**
>
> Les rôles IAM pour EMRFS fournissent un isolement au niveau de l'application entre les utilisateurs de l'application. Ils ne fournissent pas d'isolement au niveau de l'hôte entre les utilisateurs sur l'hôte. Tout utilisateur ayant accès au cluster peut contourner l'isolement pour assumer l'un de ces rôles.

Lorsqu'une application de cluster envoie une demande à Amazon S3 via EMRFS, EMRFS évalue les mappages de rôle dans l'ordre descendant dans lequel ils s'affichent dans la configuration de sécurité. Si une demande envoyée via EMRFS ne correspond à aucun identifiant, EMRFS bascule à l'aide du Rôle

de service pour les instances EC2 de cluster. Pour cette raison, nous recommandons que les stratégies attachées à ce rôle limitent les autorisations pour Amazon S3. Pour plus d'informations, consultez Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219),

Configure Roles

Avant de définir une configuration de sécurité avec des rôles IAM pour EMRFS, planifiez et créez les rôles et les stratégies d'autorisation à attacher aux rôles. Pour plus d'informations, voir Comment fonctionnent les rôles pour les instances EC2 ? dans le IAM Guide de l'utilisateur. Lorsque vous créez des politiques d'autorisations, nous vous recommandons de commencer par la politique gérée associée au rôle DME par défaut pour EC2, puis de modifier cette politique conformément à vos exigences. Le rôle par défaut est `EMR_EC2_DefaultRole`et la stratégie gérée par défaut à modifier est `AmazonElasticMapReduceforEC2Role`. Pour plus d'informations, voir Service Role for Cluster EC2 Instances (EC2 Instance Profile) (p. 219).

Updating Trust Policies for Assume Role Permissions

Chaque rôle qu'EMRFS utilise doit disposer d'une stratégie d'approbation qui permet au rôle EMR du cluster pour EC2 de l'assumer. De même, le rôle EMR du cluster pour EC2 doit disposer d'une stratégie d'approbation qui permet aux rôles EMRFS de l'assumer.

L'exemple de stratégie d'approbation suivant est attaché à des rôles pour EMRFS. L'instruction autorise le rôle EMR par défaut pour qu'EC2 assume le rôle. Par exemple, si vous avez deux rôles EMRFS fictifs, `EMRFSRole_First` et `EMRFSRole_Second`, cette déclaration de stratégie est ajoutée à la stratégie d'approbation des stratégies pour chacun d'entre eux.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::AWSAcctID:role/EMR_EC2_DefaultRole"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

De plus, l'exemple de déclaration de stratégie d'approbation suivant est ajouté à `EMR_EC2_DefaultRole` pour autoriser les deux rôles EMRFS fictifs à l'assumer.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": ["arn:aws:iam::AWSAcctID:role/EMRFSRole_First",
 "arn:aws:iam::AWSAcctID:role/EMRFSRole_Second"]
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Pour mettre à jour la stratégie d'approbation d'un rôle IAM

Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.

1. Choisissez Rôles, saisissez le nom du rôle dans Search (Rechercher), puis sélectionnez Role name (Nom de rôle).
2. Choisissez Relations d'approbation, Modifier la relation d'approbation.
3. Ajouter une instruction d'approbation en fonction du Document de stratégie, conformément aux directives ci-dessus, puis choisissez Mettre à jour la stratégie de confiance.

Specifying a Role as a Key User

Si un rôle permet d'accéder à un emplacement dans Amazon S3 qui est chiffré à l'aide d'une clé principale client (CMK) AWS Key Management Service, assurez-vous que le rôle est spécifié comme utilisateur clé. Ceci donne au rôle l'autorisation d'utiliser la CMK. Pour plus d'informations, consultez Utilisation des stratégies de clé dans le AWS Key Management Service Developer Guide.

Set Up a Security Configuration with IAM Roles for EMRFS

> **Important**
>
> Si aucun des rôles IAM pour EMRFS que vous spécifiez ne s'applique, EMRFS bascule vers le rôle EMR pour EC2. Pensez à personnaliser ce rôle pour limiter les autorisations à Amazon S3 en fonction des besoins de votre application, puis spécifiez ce rôle personnalisé au lieu de `EMR_EC2_DefaultRole` lorsque vous créez un cluster. Pour de plus amples informations, veuillez consulter Customize IAM Roles (p. 231) et Specify Custom IAM Roles When You Create a Cluster (p. 232).

Pour spécifier les rôles IAM pour les demandes EMRFS dans Amazon S3 à l'aide de la console

1. Créez une configuration de sécurité qui spécifie les mappages de rôle :
   a. Dans la console Amazon EMR, sélectionnez Security configurations (Configurations de sécurité), Create (Créer).
   b. Dans Name (Nom), saisissez un nom pour la configuration de sécurité. Ce nom est utilisé pour spécifier la configuration de sécurité lorsque vous créez un cluster.
   c. Choisissez Utiliser des rôles IAM pour les demandes EMRFS à Amazon S3.
   d. Sélectionnez un Rôle IAM à appliquer, puis, sous Base pour l'accès, sélectionnez un type d'identifiant (Users (Utilisateurs), Groups (Groupes) ou S3 prefixes (Préfixes S3)) dans la liste et saisissez les identifiants correspondants. Si vous utilisez plusieurs identifiants, séparez-les par une virgule sans insérer d'espace. Pour plus d'informations sur chaque type d'identifiant, consultez la JSON configuration reference (p. 237) ci-dessous.
   e. Choisissez Add role (Ajouter un rôle) pour configurer des mappages de rôle supplémentaires, comme décrit à l'étape précédente.
   f. Définissez les autres options de la configuration de sécurité selon vos besoins et choisissez Create (Créer). Pour plus d'informations, consultez Création d'une configuration de sécurité (p. 179),
2. Spécifiez la configuration de sécurité créée précédemment lorsque vous créez un cluster. Pour plus d'informations, consultez Spécification d'une configuration de sécurité pour un cluster (p. 196),

Pour spécifier les rôles IAM pour les demandes EMRFS à Amazon S3 à l'aide de l'AWS CLI

1. Utilisez la commande `aws emr create-security-configuration`, en spécifiant un nom pour la configuration de sécurité et les détails de la configuration de sécurité au format JSON.

   La commande exemple illustrée ci-dessous crée une configuration de sécurité avec le nom `EMRFS_Roles_Security_Configuration`. Il est basé sur une structure JSON dans le fichier `MyEmrfsSecConfig.json`, qui est enregistré dans le même répertoire où la commande est exécutée.

```
aws emr create-security-configuration --name EMRFS_Roles_Security_Configuration --
security-configuration file://MyEmrFsSecConfig.json.
```

Utilisez les directives suivantes pour la structure du `MyEmrFsSecConfig.json` fichier. Vous pouvez spécifier cette structure en même temps que les structures pour d'autres options de la configuration de sécurité. Pour plus d'informations, consultez Création d'une configuration de sécurité (p. 179),

Voici un exemple d'extrait JSON permettant de spécifier des rôles IAM personnalisés pour EMRFS dans une configuration de sécurité. Cet exemple illustre des mappages de rôles pour les trois différents types d'identifiant, suivis par une référence de paramètre.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Paramètre	Description
`"AuthorizationConfiguration":`	Obligatoire.
`"EmrFsConfiguration":`	Obligatoire. Contient les mappages de rôle.
`"RoleMappings":`	Obligatoire. Contient une ou plusieurs définitions de mappage de rôle. Les mappages de rôle sont évalués dans l'ordre descendant dans lequel ils s'affichent. Si un mappage de rôle est évalué comme true pour un appel EMRFS de données dans Amazon S3, aucun autre mappage de rôle n'est évalué et EMRFS utilise le rôle IAM spécifié pour la demande. Les mappages de rôle comprennent les paramètres obligatoires suivants :
`"Role":`	Spécifie l'identifiant ARN d'un rôle IAM au format `arn:aws:iam::account-id:role/role-name`. Il s'agit du rôle IAM assumé par Amazon EMR si la demande EMRFS à Amazon S3 correspond à l'un des `Identifiers` spécifiés.

Paramètre	Description
`"IdentifierType":`	Peut avoir l'une des valeurs suivantes : • `"User"` spécifie que les identifiants correspondent à un ou plusieurs utilisateurs Hadoop qui peuvent être des utilisateurs de compte Linux ou des principaux Kerberos. Lorsque la demande EMRFS provient du ou des utilisateurs spécifiés, le rôle IAM est assumé. • `"Prefix"` indique que l'identifiant correspond à un emplacement Amazon S3. Le rôle IAM est assumé pour les appels vers le ou les emplacements avec les préfixes spécifiés. Par exemple, le préfixe `s3://mybucket/` correspond à `s3://mybucket/mydir` et à `s3://mybucket/yetanotherdir`. • `"Group"` indique que les identifiants correspondent à un ou plusieurs groupes Hadoop. Le rôle IAM est assumé lorsque la demande provient d'un utilisateur du ou des groupes spécifiés.
`"Identifiers":`	Spécifie un ou plusieurs identifiants du type approprié. Séparez les identifiants par des virgules sans espace.

2. Utilisez la commande `aws emr create-cluster` pour créer un cluster et spécifiez la configuration de sécurité créée à l'étape précédente.

   L'exemple suivant crée un cluster avec les principales applications Hadoop par défaut installées. Le cluster utilise la configuration de sécurité créée précédemment en tant que `EMRFS_Roles_Security_Configuration` et utilise également un rôle EMR pour EC2, `EC2_Role_EMR_Restrict_S3`, qui est spécifié à l'aide de l'argument `InstanceProfile` du paramètre `--ec2-attributes`.

   > **Note**
   >
   > Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name MyEmrFsS3RolesCluster \
--release-label emr-5.30.1 --ec2-attributes
 InstanceProfile=EC2_Role_EMR_Restrict_S3,KeyName=MyKey \
--instance-type m5.xlarge --instance-count 3 \
--security-configuration EMRFS_Roles_Security_Configuration
```

Use Resource-Based Policies for Amazon EMR Access to AWS Glue Data Catalog

Si vous utilisez AWS Glue avec Hive, Spark ou Presto dans Amazon EMR, AWS Glue prend en charge les stratégies basées sur des ressources pour contrôler l'accès aux ressources Data Catalog. Ces ressources comprennent des bases de données, des tableaux, des connexions et des fonctions définies

par l'utilisateur. Pour plus d'informations, consultez Stratégies de ressources AWS Glue dans le AWS Glue Developer Guide.

Lorsque vous utilisez des stratégies basées sur des ressources pour limiter l'accès à AWS Glue de l'intérieur de Amazon EMR, le mandataire que vous spécifiez dans la stratégie de permissions doit être le rôle ARN associé au profil d'instance EC2 qui est spécifié lorsqu'un cluster est créé. Par exemple, pour une stratégie basée sur des ressources attachée à un catalogue, vous pouvez spécifier l'ARN de rôle pour le Rôle de service pour les instances EC2 de cluster par défaut, *`EMR_EC2_DefaultRole`*, comme `Principal` en utilisant le format indiqué dans l'exemple suivant :

```
arn:aws:iam::acct-id:role/EMR_EC2_DefaultRole
```

`acct-id` peut être différent de l'ID de compte AWS Glue. Cela permet un accès à partir de clusters EMR dans différents comptes. Vous pouvez spécifier plusieurs mandataires, chacun d'un compte différent.

Use IAM Roles with Applications That Call AWS Services Directly

Les applications qui s'exécutent sur les instances EC2 d'un cluster peuvent utiliser le profil d'instance EC2 afin d'obtenir des informations d'identification de sécurité temporaires lors de l'appel de services AWS.

Les versions de Hadoop disponibles avec la version Amazon EMR 2.3.0 et les versions ultérieures ont déjà été mises à jour pour utiliser des rôles IAM. Si votre application s'exécute strictement sur l'architecture Hadoop et n'appelle directement aucun service dans AWS, elle devrait fonctionner avec des rôles IAM sans modification.

Si votre application appelle des services directement dans AWS, vous devez la mettre à jour pour tirer parti des rôles IAM. Ainsi, au lieu d'obtenir des informations d'identification du compte depuis `/etc/hadoop/conf/core-site.xml` sur les instances EC2 dans le cluster, votre application utilise un SDK pour accéder aux ressources à l'aide de rôles IAM, ou appelle les métadonnées d'instance EC2 pour obtenir les informations d'identification temporaires.

Pour accéder aux ressources AWS avec des rôles IAM à l'aide d'un kit SDK

- Les rubriques suivantes montrent comment utiliser plusieurs des kits SDK AWS pour accéder aux informations d'identification temporaires à l'aide de rôles IAM. Chaque rubrique commence avec une version d'une application qui n'utilise pas de rôles IAM et puis vous guide à travers le processus de conversion de cette application pour utiliser des rôles IAM.
  - Using IAM Roles for Amazon EC2 Instances with the SDK for Java in the AWS SDK for Java Developer Guide
  - Using IAM Roles for Amazon EC2 Instances with the SDK for .NET in the Kit AWS SDK pour .NET Developer Guide
  - Using IAM Roles for Amazon EC2 Instances with the SDK for PHP in the AWS SDK pour PHP Developer Guide
  - Using IAM Roles for Amazon EC2 Instances with the SDK for Ruby in the Kit SDK AWS pour Ruby Developer Guide

Pour obtenir des informations d'identification temporaires à partir de métadonnées d'instance EC2

- Appelez l'URL suivante à partir d'une instance EC2 en cours d'exécution avec le rôle IAM spécifié, qui renvoie les informations d'identification de sécurité temporaires associées (AccessKeyId, SecretAccessKey, SessionToken et Expiration). L'exemple suivant utilise le profil d'instance par défaut pour Amazon EMR, `EMR_EC2_DefaultRole`.

```
GET http://169.254.169.254/latest/meta-data/iam/security-
credentials/EMR_EC2_DefaultRole
```

Pour plus d'informations sur l'écriture d'applications qui utilisent des rôles IAM, consultez Accorder l'accès aux ressources AWS pour les applications qui s'exécutent sur des instances Amazon EC2.

Pour plus d'informations sur les informations d'identification temporaires de sécurité, consultez Utilisations d'informations d'identification de sécurité temporaires dans le Using Temporary Security Credentials.

Allow Users and Groups to Create and Modify Roles

Les mandataires IAM (utilisateurs et groupes) qui créent, modifient, et spécifient les rôles pour un cluster, y compris les rôles par défaut, doivent être autorisés à effectuer les actions suivantes. Pour en savoir plus sur chaque action, consultez Actions dans le IAM API Reference.

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:AddRoleToInstanceProfile`
- `iam:ListRoles`
- `iam:GetPolicy`
- `iam:GetInstanceProfile`
- `iam:GetPolicyVersion`
- `iam:AttachRolePolicy`
- `iam:PassRole`

L'autorisation `iam:PassRole` permet la création de cluster. Les autorisations restantes autorisent la création des rôles par défaut.

Exemples de stratégies basées sur l'identité Amazon EMR

Par défaut, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou à modifier les ressources Amazon EMR. Ils ne peuvent pas non plus exécuter des tâches à l'aide de AWS Management Console, AWS CLI ou de l'API AWS. Un administrateur IAM doit créer des stratégies IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour apprendre à créer une stratégie basée sur l'identité IAM à l'aide de ces exemples de document de stratégie JSON, consultez Création de stratégies dans l'onglet JSON dans le IAM Guide de l'utilisateur.

Rubriques

- Bonnes pratiques en matière de stratégies pour Amazon EMR (p. 241)
- Autoriser les utilisateurs à afficher leurs propres autorisations (p. 241)
- Stratégies gérées Amazon EMR (p. 242)
- Stratégies IAM pour l'accès basé sur des balises aux clusters et aux bloc-notes EMR (p. 244)
- Refus de l’action ModifyInstanceGroup (p. 253)

Bonnes pratiques en matière de stratégies pour Amazon EMR

Les stratégies basées sur l'identité sont très puissantes. Elles déterminent si une personne peut créer, consulter ou supprimer des ressources Amazon EMR dans votre compte. Ces actions peuvent entraîner des frais pour votre compte AWS. Lorsque vous créez ou modifiez des stratégies basées sur l'identité, suivez ces instructions et recommandations :

- Commencer à utiliser des stratégies gérées AWS – Pour commencer à utiliser Amazon EMR rapidement, utilisez les politiques gérées AWS pour accorder à vos employés les autorisations dont ils ont besoin. Ces stratégies sont déjà disponibles dans votre compte et sont gérées et mises à jour par AWS. Pour de plus amples informations, veuillez consulter la section Mise en route avec les autorisations à l'aide des stratégies gérées AWS dans le IAM Guide de l'utilisateur et dans Stratégies gérées Amazon EMR (p. 242).
- Accorder le privilège le plus faible – Lorsque vous créez des stratégies personnalisées, accordez uniquement les autorisations requises pour exécuter une seule tâche. Commencez avec un minimum d'autorisations et accordez-en d'autres si nécessaire. Cette méthode est plus sûre que de commencer avec des autorisations trop permissives et d'essayer de les restreindre plus tard. Pour plus d'informations, consultez Accorder le privilège le plus faible dans le IAM Guide de l'utilisateur.
- Activer MFA pour les opérations sensibles – Pour plus de sécurité, obligez les utilisateurs IAM à utiliser l'authentification multi-facteurs (MFA) pour accéder à des ressources ou à des opérations d'API sensibles. Pour plus d'informations, consultez Utilisation de Multi-Factor Authentication (MFA) dans AWS dans le IAM Guide de l'utilisateur.
- Utiliser des conditions de stratégie pour une plus grande sécurité – Tant que cela reste pratique pour vous, définissez les conditions dans lesquelles vos stratégies basées sur l'identité autorisent l'accès à une ressource. Par exemple, vous pouvez rédiger les conditions pour spécifier une plage d'adresses IP autorisées d'où peut provenir une demande. Vous pouvez également écrire des conditions pour autoriser les requêtes uniquement à une date ou dans une plage de temps spécifiée, ou pour imposer l'utilisation de SSL ou de MFA. Pour plus d'informations, consultez IAM Éléments de stratégie JSON : État dans le IAM Guide de l'utilisateur.

Autoriser les utilisateurs à afficher leurs propres autorisations

Cet exemple montre comment créer une stratégie qui permet aux utilisateurs IAM d'afficher les stratégies en ligne et gérées attachées à leur identité d'utilisateur. Cette stratégie inclut les autorisations nécessaires pour réaliser cette action sur la console ou par programmation à l'aide de l'AWS CLI ou de l'API AWS.

```
{
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "ViewOwnUserInfo",
               "Effect": "Allow",
               "Action": [
                   "iam:GetUserPolicy",
                   "iam:ListGroupsForUser",
                   "iam:ListAttachedUserPolicies",
                   "iam:ListUserPolicies",
                   "iam:GetUser"
               ],
               "Resource": [
                   "arn:aws:iam::*:user/${aws:username}"
               ]
           },
           {
               "Sid": "NavigateInConsole",
               "Effect": "Allow",
               "Action": [
```

```
                    "iam:GetGroupPolicy",
                    "iam:GetPolicyVersion",
                    "iam:GetPolicy",
                    "iam:ListAttachedGroupPolicies",
                    "iam:ListGroupPolicies",
                    "iam:ListPolicyVersions",
                    "iam:ListPolicies",
                    "iam:ListUsers"
                ],
                "Resource": "*"
            }
        ]
    }
```

Stratégies gérées Amazon EMR

Le moyen le plus simple d'accorder un accès total ou un accès en lecture seule à des actions Amazon EMR requises consiste à utiliser les stratégies IAM gérées pour Amazon EMR. Les stratégies gérées offrent l'avantage de mises à jour automatiques dès que les exigences d'autorisations varient. Si vous utilisez des stratégies en ligne, il se peut que vous rencontriez des erreurs d'autorisation.

Ces stratégies incluent non seulement des actions pour Amazon EMR, mais elles comprennent également des actions pour Amazon EC2, Amazon S3 et Amazon CloudWatch qu'Amazon EMR utilise pour exécuter des actions comme lancer des instances, écrire des fichiers journaux et gérer des tâches Hadoop.

Les utilisateurs du cluster ont besoin d'autorisations pour que les applications puissent transmettre les rôles de service pour Amazon EMR en leur nom. La stratégie d'autorisations `AmazonElasticMapReduceFullAccess`, qui est la stratégie gérée par défaut permettant aux utilisateurs d'avoir des autorisations complètes pour Amazon EMR, inclut une instruction qui accorde les autorisations `iam:PassRole` pour toutes les ressources. Cette instruction autorise l'utilisateur à transmettre un rôle quelconque aux autres services AWS afin qu'Amazon EMR puisse interagir avec ces services au nom de l'utilisateur.

Pour mettre en œuvre une stratégie plus restrictive, attachez une stratégie en ligne aux utilisateurs ou groupes appropriés, qui accorde l'autorisation `iam:PassRole` uniquement pour les rôles spécifiques à Amazon EMR. L'exemple suivant illustre une instruction qui accorde l'autorisation `iam:PassRole` uniquement pour les rôles Amazon EMR par défaut : `EMR_DefaultRole`, `EMR_EC2_DefaultRole` et `EMR_AutoScalingDefaultRole`. Si vous utilisez des rôles personnalisés, remplacez les noms de rôles par défaut par vos noms de rôles personnalisés.

```
{
    "Action": "iam:PassRole",
    "Effect": "Allow",
    "Resource": [
        "arn:aws:iam::*:role/EMR_DefaultRole",
        "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
        "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
        "arn:aws:iam::*:role/EMR_Notebooks_DefaultRole"
    ]
}
```

Pour créer des stratégies personnalisées, nous vous recommandons de commencer avec des stratégies gérées, puis de les modifier selon vos besoins.

Pour plus d'informations sur la façon de lier des stratégies aux utilisateurs IAM (mandataires), consultez Utilisation de stratégies gérées à l'aide de la AWS Management Console dans le IAM Guide de l'utilisateur.

Stratégie gérée IAM pour un accès complet

Pour accorder toutes les actions requises pour Amazon EMR, attachez la stratégie gérée `AmazonElasticMapReduceFullAccess`. Le contenu de cette déclaration de stratégie est présenté ci-dessous. Il révèle toutes les actions dont Amazon EMR a besoin pour les autres services.

Le contenu de la version 6 de cette stratégie est présenté ci-dessous. Étant donné que la stratégie AmazonElasticMapReduceFullAccess est automatiquement mise à jour, la stratégie affichée ici peut ne pas être à jour. Utilisez AWS Management Console pour afficher la stratégie actuelle.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudwatch:*",
                "cloudformation:CreateStack",
                "cloudformation:DescribeStackEvents",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:CancelSpotInstanceRequests",
                "ec2:CreateRoute",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DeleteRoute",
                "ec2:DeleteTags",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs",
                "ec2:DescribeRouteTables",
                "ec2:DescribeNetworkAcls",
                "ec2:CreateVpcEndpoint",
                "ec2:ModifyImageAttribute",
                "ec2:ModifyInstanceAttribute",
                "ec2:RequestSpotInstances",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "elasticmapreduce:*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:ListRoles",
                "iam:PassRole",
                "kms:List*",
                "s3:*",
                "sdb:*",
                "support:CreateCase",
                "support:DescribeServices",
                "support:DescribeSeverityLevels"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "elasticmapreduce.amazonaws.com",
                        "elasticmapreduce.amazonaws.com.cn"
                    ]
                }
            }
        }
    ]
}
```

Note

L'action `ec2:TerminateInstances` permet à l'utilisateur IAM de mettre hors service n'importe laquelle des instances Amazon EC2 associées au compte IAM, même celles qui ne font pas partie d'un cluster EMR.

Stratégie IAM gérée pour un accès en lecture seule

Pour accorder les privilèges en lecture seule à Amazon EMR, attachez la stratégie gérée AmazonElasticMapReduceReadOnlyAccess. Le contenu de cette déclaration de stratégie est présenté ci-dessous. Les caractères génériques de l'élément `elasticmapreduce` indiquent que seules les actions commençant par des chaînes spécifiées sont autorisées. Gardez à l'esprit que, si cette stratégie n'empêche pas explicitement certaines actions, une autre déclaration de stratégie peut toutefois être utilisée pour accorder l'accès aux actions spécifiées.

Note

Étant donné que la stratégie AmazonElasticMapReduceReadOnlyAccess est automatiquement mise à jour, la stratégie affichée ici peut ne pas être à jour. Utilisez AWS Management Console pour afficher la stratégie actuelle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:Describe*",
        "elasticmapreduce:List*",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole"
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sdb:Select",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```

Stratégies IAM pour l'accès basé sur des balises aux clusters et aux bloc-notes EMR

Vous pouvez utiliser des conditions dans votre stratégie basée sur les identités pour contrôler l'accès aux clusters et blocs-notes EMR en fonction des balises.

Pour de plus amples informations sur l'ajout de balises à des clusters, veuillez consulter Balisage de clusters EMR. Pour de plus amples informations sur l'utilisation de ces clés de condition, veuillez consulter Clés de condition (p. 212).

Les exemples suivants illustrent différents scénarios et différentes façons d'utiliser des opérateurs de condition avec des clés de condition Amazon EMR. Ces déclarations de stratégie IAM sont conçues uniquement à des fins de démonstration et ne doivent pas être utilisées dans des environnements de production. Il existe plusieurs façons de combiner des instructions de stratégie pour accorder et refuser des autorisations selon vos besoins. Pour plus d'informations sur la planification et la vérification des stratégies IAM, consultez le IAM Guide de l'utilisateur.

> **Important**
>
> Le refus explicite d'autoriser l'attribution de balises doit être pris en considération. Cela empêche les utilisateurs de baliser une ressource et donc d'accorder eux-mêmes des autorisations que vous n'aviez pas l'intention d'accorder à. Si les actions de balisage d'une ressource ne sont pas refusées, un utilisateur peut modifier les balises et contourner l'intention des stratégies basées sur les balises. Pour obtenir un exemple d'une stratégie qui refuse les actions de balisage, consultez Refuser l'accès pour ajouter et supprimer des balises (p. 247).

Exemple de déclarations de stratégie basées sur les identités pour les clusters

Les exemples ci-dessous illustrent les stratégies d'autorisation basées sur une identité qui sont utilisées pour contrôler les actions autorisées avec les clusters EMR.

> **Important**
>
> Le `ModifyInstanceGroup` action dans Amazon EMR ne nécessite pas que vous spécifiez un ID de cluster. Pour cette raison, le refus de cette action en fonction des balises de cluster nécessite une considération supplémentaire. Pour plus d'informations, consultez Refus de l'action ModifyInstanceGroup (p. 253),

Rubriques

- Actions autorisées uniquement sur des clusters avec des valeurs de balises spécifiques (p. 245)
- Balisage de cluster requis lors de la création d'un cluster (p. 246)
- Refuser l'accès pour ajouter et supprimer des balises (p. 247)
- Actions autorisées sur des clusters avec une balise spécifique, quelle que soit la valeur de la balise (p. 247)
- Obligation pour les utilisateurs d'ajouter des balises lors de la création d'un cluster (p. 248)

Actions autorisées uniquement sur des clusters avec des valeurs de balises spécifiques

Les exemples ci-dessous illustrent une stratégie qui permet à un utilisateur d'effectuer des actions sur la base de la balise du cluster *`department`* avec la valeur *`dev`* et qui permet également à un utilisateur de baliser des clusters en utilisant cette même balise. L'exemple de stratégie finale montre comment refuser les privilèges permettant d'attribuer aux clusters EMR des balises autres que cette même balise.

Dans l'exemple de stratégie suivant, l' `StringEquals` essaie de faire correspondre l'opérateur de condition *`dev`* avec la valeur de la balise *`department`*. Si la balise *`department`* n'a pas été ajouté au cluster ou ne contient pas la valeur *`dev`*, la stratégie ne s'applique pas et les actions ne sont pas autorisées par cette stratégie. Si aucune autre déclaration de stratégie n'autorise ces actions, l'utilisateur peut uniquement utiliser des clusters ayant cette balise avec cette valeur.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt12345678901234",
```

```
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:TerminateJobFlows",
        "elasticmapreduce:SetTerminationProtection",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:DescribeStep"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": "dev"
        }
      }
    }
  ]
}
```

Vous pouvez également spécifier plusieurs valeurs de balise à l'aide d'un opérateur de condition. Par exemple, pour autoriser toutes les actions sur des clusters où la balise *department* a la valeur *dev* ou *test*, vous pouvez remplacer le bloc de condition dans l'exemple précédent avec les éléments suivants.

```
            "Condition": {
              "StringEquals": {
                "elasticmapreduce:ResourceTag/department":["dev", "test"]
              }
            }
```

Balisage de cluster requis lors de la création d'un cluster

Comme dans l'exemple précédent, l'exemple de stratégie suivant recherche la même balise correspondante : la valeur *dev* pour la balise *department*. Dans ce cas, toutefois, la clé de condition `RequestTag` spécifie que la stratégie s'applique lors de la création de la balise, ainsi l'utilisateur doit créer une balise qui correspond à la valeur spécifiée.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1479334524000",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:RunJobFlow",
        "iam:PassRole"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:RequestTag/department": "dev"
        }
      }
    }
  ]
```

```
}
```

Refuser l'accès pour ajouter et supprimer des balises

Dans l'exemple suivant, les actions EMR qui permettent l'ajout et la suppression des balises est associée à un opérateur `StringNotEquals` spécifiant la balise `dev` que nous avons vue dans les exemples précédents. L'effet de cette stratégie consiste à refuser à un utilisateur l'autorisation d'ajouter ou de supprimer des balises sur des clusters EMR qui comportent une balise *`department`* comportant la valeur *`dev`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
    "Effect": "Deny",
    "Action": [
      "elasticmapreduce:AddTags",
      "elasticmapreduce:RemoveTags"
      ],
      "Condition": {
        "StringNotEquals": {
          "elasticmapreduce:ResourceTag/department": "dev"
        }
      },
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Actions autorisées sur des clusters avec une balise spécifique, quelle que soit la valeur de la balise

Vous pouvez également autoriser des actions uniquement sur des clusters ayant une balise spécifique, quelle que soit la valeur de la balise. Pour cela, vous pouvez utiliser l'opérateur `Null`. Pour plus d'informations, consultez la page Opérateur de condition pour vérifier l'existence de clés de condition dans le IAM Guide de l'utilisateur. Par exemple, pour autoriser des actions uniquement sur des clusters EMR qui ont la balise *`department`*, quelle que soit sa valeur, vous pouvez remplacer les blocs de condition de l'exemple précédent par le suivant. L'opérateur `Null` recherche la balise *`department`* sur un cluster EMR. Si la balise existe, l'instruction `Null` a la valeur false, correspondant à la condition spécifiée dans cette déclaration de stratégie, et les actions appropriées sont autorisées.

```
"Condition": {
  "Null": {
    "elasticmapreduce:ResourceTag/department":"false"
  }
}
```

La déclaration de stratégie suivante permet à un utilisateur de créer un cluster EMR uniquement s'il a une balise *`department`*, quelle que soit la valeur de cette dernière.

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

```
    {
    "Action": [
      "elasticmapreduce:RunJobFlow",
      "iam:PassRole"
      ],
    "Condition": {
      "Null": {
        "elasticmapreduce:RequestTag/department": "false"
        }
      },
    "Effect": "Allow",
    "Resource": [
      "*"
      ]
    }
  ]
}
```

Obligation pour les utilisateurs d'ajouter des balises lors de la création d'un cluster

La déclaration de stratégie suivante permet à un utilisateur de créer un cluster EMR uniquement s'il a une balise *department* qui contient la valeur *dev* lors de sa création.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:RunJobFlow",
                "iam:PassRole"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/department": "dev"
                }
            },
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Exemple de déclarations de stratégie basées sur les identités pour Blocs-notes EMR

Les exemples de déclarations de stratégies IAM dans cette section illustrent les scénarios courants pour l'utilisation de clés afin de limiter les actions autorisées en utilisant Blocs-notes EMR. Dans la mesure où aucune autre stratégie associée au mandataire (utilisateur) autorise les actions, les clés de contexte de condition limitent les actions autorisées comme indiqué.

Example – Autoriser l’accès uniquement aux blocs-notes qu’un utilisateur crée en fonction du balisage

L'exemple de déclaration de stratégie ci-dessous, lorsqu'il est attaché à un rôle ou à un utilisateur, permet à l'utilisateur IAM de travailler uniquement avec les blocs-notes qu'ils ont créés. Cette déclaration de stratégie utilise la balise par défaut appliquée lorsqu'un bloc-notes est créé.

Dans l'exemple, le `StringEquals` l'opérateur de condition essaie de faire correspondre une variable représentant les utilisateurs actuels IAM ID utilisateur (`{aws:userId}`) avec la valeur de la balise `creatorUserID`. Si la balise `creatorUserID` n'a pas été ajouté au bloc-notes ou ne contient pas la valeur de l'ID de l'utilisateur actuel, la stratégie ne s'applique pas et les actions ne sont pas autorisées par cette stratégie. Si aucune autre déclaration de stratégie n'autorise ces actions, l'utilisateur peut uniquement utiliser des blocs-notes ayant cette balise avec cette valeur.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
                }
            }
        }
    ]
}
```

Example –Exiger le balisage de bloc-notes lorsqu'un bloc-notes est créé

Dans cet exemple, la clé de contexte `RequestTag` est utilisée. L'action `CreateEditor` est autorisée uniquement si l'utilisateur n'a pas modifié ou supprimé la balise `creatorUserID` est ajoutée par défaut. La variable ${aws:userId}, spécifie l'ID d'utilisateur de l'utilisateur actuellement actif, ce qui est la valeur par défaut de la balise.

La déclaration de stratégie peut être utilisée pour aider à garantir que les utilisateurs ne suppriment pas la balise `createUserId` ou ne modifient sa valeur.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/creatorUserId": "${aws:userid}"
                }
            }
        }
    ]
}
```

Cet exemple exige que l'utilisateur crée le cluster avec une balise ayant la chaîne de clé `dept` et une valeur définie sur l'une des valeurs suivantes : `datascience`, `analytics`, `operations`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/dept": [
                        "datascience",
                        "analytics",
                        "operations"
                    ]
                }
            }
        }
    ]
}
```

Example –Limitez la création de bloc-notes aux clusters balisés et exigez des balises de blocs-notes

Cet exemple autorise la création de bloc-notes uniquement si le bloc-notes est créé avec une balise qui dispose de la chaîne de clé `owner` définie sur l'une des valeurs spécifiées. De plus, le bloc-notes peut être créé uniquement si le cluster dispose d'une balise avec la chaîne de clé `department` définie sur l'une des valeurs spécifiées.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/owner": [
                        "owner1",
                        "owner2",
                        "owner3"
                    ],
                    "elasticmapreduce:ResourceTag/department": [
                        "dep1",
                        "dep3"
                    ]
                }
            }
        }
    ]
}
```

Example –Limitez la possibilité de démarrer un bloc-notes basé sur des balises

Cet exemple limite la capacité de lancer des blocs-notes uniquement aux blocs-notes dotés d'une balise avec la chaîne de clé `owner` définie sur une des valeurs spécifiées. Étant donné que l'élément `Resource` est utilisé pour spécifier uniquement l'`editor`, la condition ne s'applique pas au cluster et n'a pas besoin d'être balisée.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "owner1",
                        "owner2"
                    ]
                }
            }
        }
    ]
}
```

Cet exemple est similaire à un exemple précédent. Toutefois, la limite s'applique uniquement aux clusters balisés, pas aux blocs-notes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "dep1",
                        "dep3"
                    ]
                }
            }
        }
    ]
}
```

Cet exemple utilise un autre ensemble de bloc-notes et de balises de clusters. Il permet à un bloc-notes de se lancer uniquement si :

- Le bloc-notes a une balise avec la chaîne de clé `owner` définie sur l'une des valeurs spécifiées.

  —$and—

- Le cluster a une balise avec la chaîne de clé `department` définie sur l'une des valeurs spécifiées

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
```

```
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "datascience",
                        "analytics"
                    ]
                }
            }
        }
    ]
}
```

Example –Limitez la possibilité d'ouvrir l'éditeur de bloc-notes basé sur des balises

Cet exemple autorise l'éditeur de bloc-notes à s'ouvrir uniquement si :

- Le bloc-notes a une balise avec la chaîne de clé `owner` définie sur l'une des valeurs spécifiées.

  —$and—
- Le cluster a une balise avec la chaîne de clé `department` définie sur l'une des valeurs spécifiées

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
```

```
                        "datascience",
                        "analytics"
                    ]
                }
            }
        }
    ]
}
```

Refus de l'action ModifyInstanceGroup

Le Groupes d'instances de modification action dans Amazon EMR ne nécessite pas que vous fournissiez un ID de cluster avec l'action. Au lieu de cela, vous pouvez spécifier uniquement un ID de groupe d'instances. Pour cette raison, une stratégie de refus apparemment simple pour cette action basée sur l'ID de cluster ou une balise de cluster peut ne pas avoir l'effet prévu. Examinons l'exemple de stratégie suivant.

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
        }
    ]

}
```

Si un utilisateur avec cette stratégie attachée effectue un `ModifyInstanceGroup` et spécifie uniquement l'ID du groupe d'instances, la stratégie n'applique pas. Étant donné que l'action est autorisée sur toutes les autres ressources, l'action aboutit.

Une solution à ce problème consiste à attacher une déclaration de stratégie à l'identité qui utilise un Pas de ressource pour refuser tout `ModifyInstanceGroup` action émise sans ID de cluster. L'exemple de stratégie suivant ajoute une telle instruction de refus afin que `ModifyInstanceGroups` à moins qu'un ID de cluster ne soit spécifié. Puisqu'une identité doit spécifier un ID de cluster avec l'action , les instructions de refus basées sur l'ID de cluster sont donc effectives.

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

```
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "NotResource": "arn:*:elasticmapreduce:*:*:cluster/*"
        }
    ]

}
```

Un problème similaire existe lorsque vous souhaitez refuser le `ModifyInstanceGroups` action basée sur la valeur associée à une balise de cluster. La solution est similaire. En plus d'une instruction de refus qui spécifie la valeur de balise, vous pouvez ajouter une instruction de stratégie qui refuse le `ModifyInstanceGroup` action si la balise que vous spécifiez n'est pas présente, quelle que soit la valeur.

L'exemple suivant illustre une stratégie qui, lorsqu'elle est attachée à une identité, refuse l'identité `ModifyInstanceGroups` action sur n'importe quel cluster avec la balise `department` défini sur `dev`. Cette instruction n'est efficace qu'en raison de l'instruction de refus qui utilise le `StringNotLike` pour refuser l'action à moins que le `department` est présent.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "dev"
                }
            },
            "Effect": "Deny",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:ResourceTag/department": "?*"
                }
            },
            "Effect": "Deny",
            "Resource": "*"
        }
    ],
}
```

Authenticate to Amazon EMR Cluster Nodes

Des clients SSH peuvent utiliser une paire de clés Amazon EC2 pour authentifier des instances de cluster. Sinon, avec Amazon EMR version 5.10.0 ou version ultérieure, vous pouvez configurer Kerberos pour authentifier les utilisateurs et les connexions SSH sur le nœud principal. Pour plus d'informations, consultez Use Kerberos Authentication (p. 255),

Rubriques

- Use an Amazon EC2 Key Pair for SSH Credentials (p. 255)
- Use Kerberos Authentication (p. 255)

Use an Amazon EC2 Key Pair for SSH Credentials

Les nœuds de cluster Amazon EMR sont exécutés sur des instances Amazon EC2. Vous pouvez vous connecter aux nœuds de cluster de la même manière qu'aux instances Amazon EC2. Vous pouvez utiliser Amazon EC2 pour créer une paire de clés ou vous pouvez en importer une. Lorsque vous créez un cluster, vous pouvez spécifier la paire de clés Amazon EC2 qui sera utilisée pour les connexions SSH à toutes les instances de cluster. Vous pouvez également créer un cluster sans paire de clés. Ceci est généralement effectué avec des clusters transitoires qui sont lancés, exécutent des étapes, puis sont mis hors service automatiquement.

Le client SSH que vous utilisez pour vous connecter a besoin du fichier de clé privée associé à cette paire de clés. Il s'agit d'un fichier .pem pour les clients SSH qui utilisent Linux, Unix et macOS. Vous devez définir les autorisations de telle sorte que seul le propriétaire des clés soit autorisé à accéder au fichier. Il s'agit d'un fichier .ppk pour les clients SSH qui utilisent Windows et le fichier .ppk est généralement créé à partir du fichier .pem.

- For more information about creating an Amazon EC2 key pair, see Amazon EC2 Key Pairs in the Amazon EC2 Guide de l'utilisateur pour les instances Linux.
- For instructions about using PuTTYgen to create a .ppk file from a .pem file, see Converting Your Private Key Using PuTTYgen in the Amazon EC2 Guide de l'utilisateur pour les instances Linux.
- For more information about setting .pem file permissions and how to connect to an EMR cluster's master node using different methods—including `ssh` from Linux or macOS, PuTTY from Windows, or the AWS CLI from any supported operating system, see Connexion au nœud maître à l'aide de SSH (p. 368).

Use Kerberos Authentication

Amazon EMR version 5.10.0 et versions ultérieures prend en charge Kerberos, qui est un protocole d'authentification réseau créé par le Massachusetts Institute of Technology (MIT). Kerberos utilise le chiffrement par clé secrète pour fournir une authentification renforcée afin que les mots de passe ou les autres informations d'identification ne soient pas envoyés sur le réseau dans un format non chiffré.

Dans Kerberos, les services et les utilisateurs qui doivent s’authentifier sont connus sous le nom de principals. Des mandants existent dans un Kerberos realm. Dans le royaume, un serveur Kerberos appelé key distribution center (KDC) permet aux directeurs d’authentifier. Le KDC fait cela en émettant tickets pour l’authentification. Il gère une base de données des principaux au sein de son domaine, leurs mots de passe, ainsi que d'autres informations administratives sur chaque principal. Un KDC peut également accepter les identifiants d’authentification des mandants d’autres royaumes, ce qui est connu sous le nom de cross-realm trust. En outre, un cluster EMR peut utiliser un KDC externe pour authentifier les principes.

Un scénario courant pour établir une approbation inter-domaines ou pour utiliser un KDC externe consiste à authentifier les utilisateurs d'un domaine Active Directory. Cela permet aux utilisateurs d'accéder à un cluster EMR à l'aide de leur compte d'utilisateur de domaine lorsqu'ils utilisent SSH pour se connecter à un cluster ou utiliser les applications de Big Data.

Lorsque vous utilisez l'authentification Kerberos, Amazon EMR configure Kerberos pour les applications, les composants et les sous-systèmes qu'il installe sur le cluster afin qu'ils puissent s'authentifier les uns les autres.

> **Important**
>
> Amazon EMR ne prend pas en charge AWS Directory Service for Microsoft Active Directory (Enterprise Edition) dans une approbation inter-domaines ou comme KDC externe.

Avant de configurer Kerberos à l'aide d'Amazon EMR, nous vous recommandons de vous familiariser avec les concepts Kerberos, les services qui s'exécutent sur un KDC et les outils d'administration des services Kerberos. Pour plus d'informations, consultez la Documentation MIT Kerberos qui est publiée par le Consortium Kerberos.

Rubriques

- Supported Applications (p. 256)
- Kerberos Architecture Options (p. 257)
- Configuring Kerberos on Amazon EMR (p. 265)
- Using SSH to Connect to Kerberized Clusters (p. 274)
- Tutorial: Configure a Cluster-Dedicated KDC (p. 275)
- Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277)

Supported Applications

Dans un cluster EMR, les mandataires Kerberos sont les services d'application et les sous-systèmes de big data qui s'exécutent sur tous les nœuds de cluster. Amazon EMR peut configurer les applications et les composants répertoriés ci-dessous pour utiliser Kerberos. Chaque application a un principal utilisateur Kerberos qui lui est associé.

Amazon EMR ne prend pas en charge les approbations inter-domaines avec AWS Directory Service for Microsoft Active Directory (Enterprise Edition).

Amazon EMR configure uniquement les fonctionnalités d'authentification Kerberos en open source pour les applications et les composants répertoriés ci-dessous. Toutes les autres applications installées ne sont pas activées pour Kerberos, ce qui peut entraîner une incapacité à communiquer avec les composants activés pour Kerberos et provoquer des erreurs d'applications. Les applications et composants qui ne sont pas activés pour Kerberos ne peuvent pas s'authentifier. Les applications et composants pris en charge peuvent varier selon la version d'Amazon EMR.

Aucune interface utilisateur web hébergée sur le cluster n'est activée pour Kerberos.

- Hadoop MapReduce
- Hbase
- HCatalog
- HDFS
- Hive
  - Do not enable Hive with LDAP authentication. This may cause issues communicating with Kerberized YARN.
- Hue
  - Hue user authentication isn't set automatically and can be configured using the configuration API.
  - Hue server is Kerberized. The Hue front-end (UI) is not configured for authentication. LDAP authentication can be configured for the Hue UI.
- Livy
- Oozie

- Phoenix
- Spark
- Tez
- YARN
- Zeppelin
  - Zeppelin is only configured to use Kerberos with the Spark interpreter. It is not configured for other interpreters.
  - Zeppelin impersonation with Kerberos is not supported. All users logged in to Zeppelin use the same Zeppelin user principal to run Spark jobs and authenticate to YARN.
- Zookeeper
  - Zookeeper client is not supported.

Kerberos Architecture Options

Lorsque vous utilisez Kerberos avec Amazon EMR, vous pouvez choisir parmi les architectures répertoriées dans cette section. Quelle que soit l'architecture que vous choisissez, vous devez configurer Kerberos à l'aide des mêmes étapes. Vous créez une configuration de sécurité, vous spécifiez la configuration de sécurité et des options Kerberos propres au cluster compatibles lorsque vous créez le cluster, et vous créez des annuaires HDFS pour des utilisateurs Linux sur le cluster qui correspondent aux mandataires d'utilisateurs dans le KDC. Pour plus d'informations sur les options de configuration et les exemples de configurations pour chaque architecture, consultez Configuring Kerberos on Amazon EMR (p. 265).

Cluster-Dedicated KDC (KDC on Master Node)

Cette configuration est disponible dans les versions Amazon EMR 5.10.0 et ultérieures.

Advantages

- Amazon EMR has full ownership of the KDC.
- The KDC on the EMR cluster is independent from centralized KDC implementations such as Microsoft Active Directory or AWS Managed Microsoft AD.
- Performance impact is minimal because the KDC manages authentication only for local nodes within the cluster.
- Optionally, other Kerberized clusters can reference the KDC as an external KDC. For more information, see External KDC—Master Node on a Different Cluster (p. 261).

Considerations and Limitations

- Kerberized clusters can not authenticate to one another, so applications can not interoperate. If cluster applications need to interoperate, you must establish a cross-realm trust between clusters, or set up one cluster as the external KDC for other clusters. If a cross-realm trust is established, the KDCs must have different Kerberos realms.
- You must create Linux users on the EC2 instance of the master node that correspond to KDC user principals, along with the HDFS directories for each user.
- User principals must use an EC2 private key file and `kinit` credentials to connect to the cluster using SSH.

Cross-Realm Trust

Dans cette configuration, des mandataires (généralement des utilisateurs) provenant d'un autre domaine Kerberos authentifient auprès de composants d'application sur un cluster EMR activé pour Kerberos, qui a son propre KDC. Le KDC du nœud principal établit une relation de confiance avec un autre KDC en utilisant cross-realm principal qui existe dans les deux kdcs. Le nom et le mot de passe du mandataire correspondent exactement dans chaque KDC. Les relations d'approbation inter-domaines sont plus communes avec les implémentations Active Directory, comme illustré dans le schéma suivant. Les relations d'approbation inter-domaines avec un KDC MIT externe ou un KDC sur un autre cluster Amazon EMR sont également prises en charge.

Advantages

- The EMR cluster on which the KDC is installed maintains full ownership of the KDC.
- With Active Directory, Amazon EMR automatically creates Linux users that correspond to user principals from the KDC. You still must create HDFS directories for each user. In addition, user principals in the Active Directory domain can access Kerberized clusters using `kinit` credentials, without the EC2 private key file. This eliminates the need to share the private key file among cluster users.

- Because each cluster KDC manages authentication for the nodes in the cluster, the effects of network latency and processing overhead for a large number of nodes across clusters is minimized.

Considerations and Limitations

- If you are establishing a trust with an Active Directory realm, you must provide an Active Directory user name and password with permissions to join principals to the domain when you create the cluster.
- Cross-realm trusts cannot be established between Kerberos realms with the same name.
- Cross-realm trusts must be established explicitly. For example, if Cluster A and Cluster B both establish a cross-realm trust with a KDC, they do not inherently trust one another and their applications cannot authenticate to one another to interoperate.
- KDCs must be maintained independently and coordinated so that credentials of user principals match precisely.

External KDC

Les configurations avec un KDC externe sont prises en charge avec les versions d'Amazon EMR 5.20.0 et ultérieures.

- External KDC—MIT KDC (p. 260)
- External KDC—Master Node on a Different Cluster (p. 261)
- External KDC—Cluster KDC on a Different Cluster with Active Directory Cross-Realm Trust (p. 263)

External KDC—MIT KDC

Cette configuration permet à un ou plusieurs clusters EMR d'utiliser les mandataires définis et maintenus dans un serveur KDC MIT.

Advantages

- Managing principals is consolidated in a single KDC.
- Multiple clusters can use the same KDC in the same Kerberos realm. This allows cluster applications to interoperate and simplifies the authentication of communication between clusters as compared to a cross-realm trust.
- The master node on a Kerberized cluster does not have the performance burden associated with maintaining the KDC.

Considerations and Limitations

- You must create Linux users on the EC2 instance of each Kerberized cluster's master node that correspond to KDC user principals, along with the HDFS directories for each user.
- User principals must use an EC2 private key file and `kinit` credentials to connect to Kerberized clusters using SSH.
- Each node in Kerberized EMR clusters must have a network route to the KDC.
- Each node in Kerberized clusters places an authentication burden on the external KDC, so the configuration of the KDC affects cluster performance. When you configure the hardware of the KDC server, consider the maximum number of Amazon EMR nodes to be supported simultaneously.
- Cluster performance is dependent on the network latency between nodes in Kerberized clusters and the KDC.
- Troubleshooting can be more difficult because of interdependencies.

External KDC—Master Node on a Different Cluster

Cette configuration est presque identique à l'implémentation KDC MIT externe ci-dessus, sauf que le KDC est sur le nœud principal d'un cluster EMR. Pour de plus amples informations, veuillez consulter Cluster-Dedicated KDC (KDC on Master Node) (p. 257) et Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277).

Advantages

- Managing principals is consolidated in a single KDC.
- Multiple clusters can use the same KDC in the same Kerberos realm. This allows cluster applications an Kerberized clusters to interoperate. It also simplifies the authentication of communication between clusters as compared to a cross-realm trust.

Considerations and Limitations

- You must create Linux users on the EC2 instance of each Kerberized cluster's master node that correspond to KDC user principals, along with the HDFS directories for each user.
- User principals must use an EC2 private key file and `kinit` credentials to connect to Kerberized clusters using SSH.
- Each node in each EMR cluster must have a network route to the KDC.
- Each Amazon EMR node in Kerberized clusters places an authentication burden on the external KDC, so the configuration of the KDC affects cluster performance. When you configure the hardware of the KDC server, consider the maximum number of Amazon EMR nodes to be supported simultaneously.
- Cluster performance is dependent on the network latency between nodes in the clusters and the KDC.
- Troubleshooting can be more difficult because of interdependencies.

External KDC—Cluster KDC on a Different Cluster with Active Directory Cross-Realm Trust

Dans cette configuration, vous devez d'abord créer un cluster avec un KDC dédié au cluster qui dispose d'une relation d'approbation inter-domaines unidirectionnelle avec Active Directory. Pour voir un didacticiel détaillé, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277). Vous pouvez ensuite lancer d'autres clusters faisant référence au cluster KDC qui a la confiance comme KDC externe. Pour obtenir un exemple, veuillez consulter External Cluster KDC with Active Directory Cross-Realm Trust (p. 271). Cela permet à chaque cluster Amazon EMR qui utilise le KDC externe d'authentifier les mandataires définis et maintenus dans un domaine Microsoft Active Directory.

Advantages

- Managing principals is consolidated in the Active Directory domain.
- Amazon EMR joins the Active Directory realm, which eliminates the need to create Linux users that correspond Active Directory users. You still must create HDFS directories for each user.
- Multiple clusters can use the same KDC in the same Kerberos realm, which is different from the Active Directory realm. This allows cluster applications to interoperate.
- User principals in the Active Directory domain can access Kerberized clusters using `kinit` credentials, without the EC2 private key file. This eliminates the need to share the private key file among cluster users.

- Only one Amazon EMR master node has the burden of maintaining the KDC, and only that cluster must be created with Active Directory credentials for the cross-realm trust between the KDC and Active Directory.

Considerations and Limitations

- Each node in each EMR cluster must have a network route to the KDC and the Active Directory domain controller.
- Each Amazon EMR node places an authentication burden on the external KDC, so the configuration of the KDC affects cluster performance. When you configure the hardware of the KDC server, consider the maximum number of Amazon EMR nodes to be supported simultaneously.
- Cluster performance is dependent on the network latency between nodes in the clusters and the KDC server.
- Troubleshooting can be more difficult because of interdependencies.

Configuring Kerberos on Amazon EMR

Cette section fournit des détails de configuration et des exemples pour configurer Kerberos avec des architectures courantes. Quelle que soit l'architecture que vous choisissez, les configurations de base sont identiques et effectuées en trois étapes. Si vous utilisez un KDC externe ou si vous configurez une approbation inter-domaines, vous devez vous assurer que tous les nœuds d'un cluster disposent d'un routage de réseau vers le KDC externe, y compris la configuration des groupes de sécurité pertinents pour autoriser le trafic Kerberos entrant et sortant.

Step 1: Create a security configuration with Kerberos properties

La configuration de sécurité spécifie les détails sur le KDC Kerberos et autorise la réutilisation de la configuration de Kerberos chaque fois que vous créez un cluster. Pour créer une configuration de sécurité, utilisez la console Amazon EMR, l'AWS CLI ou l'API EMR. La configuration de sécurité peut également contenir d'autres options de sécurité, telles que le chiffrement. Pour plus d'informations sur la création de configurations de sécurité et la spécification d'une configuration de sécurité lorsque vous créez un cluster, consultez Utilisation de configurations de sécurité pour configurer la sécurité du cluster (p. 179). Pour plus d'informations sur les propriétés Kerberos dans une configuration de sécurité, consultez Kerberos Settings for Security Configurations (p. 266).

Step 2: Create a cluster and specify cluster-specific Kerberos attributes

Lorsque vous créez un cluster, spécifiez une configuration de sécurité Kerberos ainsi que des options Kerberos spécifiques au cluster. Lorsque vous utilisez la console Amazon EMR, seules les options Kerberos compatibles avec la configuration de sécurité spécifiée sont disponibles. Lorsque vous utilisez l'AWS CLI ou l'API Amazon EMR, assurez-vous de préciser les options Kerberos compatibles avec la configuration de sécurité spécifiée. Par exemple, si vous spécifiez un mot de passe de mandataire pour une approbation inter-domaines lorsque vous créez un cluster à l'aide de l'interface de ligne de commande, et la configuration de sécurité spécifiée n'est pas configurée avec les paramètres d'approbation inter-domaines, une erreur se produit. Pour plus d'informations, consultez Kerberos Settings for Clusters (p. 269),

Step 3: Configure the cluster master node

Selon les exigences de votre architecture et l'implémentation, une configuration supplémentaire sur le cluster peut être requise. Vous pouvez effectuer cette opération après l'avoir créée ou en utilisant les étapes ou les actions d'amorçage pendant le processus de création.

Pour chaque utilisateur authentifié par Kerberos qui se connecte au cluster à l'aide de SSH, vous devez vous assurer que les comptes d'utilisateur Linux qui correspondent à l'utilisateur Kerberos sont créés. Si les mandataires d'utilisateurs sont fournis par un contrôleur de domaine Active Directory, soit comme

KDC externe ou par le biais d'une approbation inter-domaines, Amazon EMR crée automatiquement des comptes d'utilisateur Linux. Si Active Directory n'est pas utilisé, vous devez créer des mandataires pour chaque utilisateur qui correspondent à leur utilisateur Linux. Pour plus d'informations, consultez Configuring a Cluster for Kerberos-Authenticated HDFS Users and SSH Connections (p. 271),

Chaque utilisateur doit également disposer d'un répertoire d'utilisateurs HDFS qui leur appartient et que vous devez créer. En outre, SSH doit être configuré avec la GSSAPI activée afin d'autoriser les connexions à partir des utilisateurs authentifiés par Kerberos. GSSAPI doit être activée sur le nœud principal et l'application SSH cliente doit être configurée pour utiliser la GSSAPI. Pour plus d'informations, consultez Configuring a Cluster for Kerberos-Authenticated HDFS Users and SSH Connections (p. 271),

Security Configuration and Cluster Settings for Kerberos on Amazon EMR

Lorsque vous créez un cluster activé pour Kerberos, vous spécifiez la configuration de sécurité avec des attributs Kerberos qui sont propres au cluster. Vous ne pouvez pas spécifier un ensemble sans l'autre, sinon une erreur se produit.

Cette rubrique fournit une vue d'ensemble des paramètres de configuration disponibles pour Kerberos lorsque vous créez une configuration de sécurité et un cluster. De plus, les exemples de l'interface de ligne de commande pour créer des clusters et des configurations de sécurité compatibles sont fournis pour les architectures courantes.

Kerberos Settings for Security Configurations

Vous pouvez créer une configuration de sécurité qui spécifie les attributs Kerberos à l'aide de la console Amazon EMR, de l'AWS CLI ou de l'API EMR. La configuration de sécurité peut également contenir d'autres options de sécurité, telles que le chiffrement. Pour plus d'informations, consultez Création d'une configuration de sécurité (p. 179),

Utilisez les références suivantes pour comprendre les paramètres de configuration de sécurité disponibles pour l'architecture Kerberos que vous choisissez. Les paramètres de console Amazon EMR sont affichés. Pour les options d'interface de ligne de commande correspondantes, consultez Spécification des paramètres Kerberos à l'aide de l'AWS CLI (p. 191) ou Configuration Examples (p. 270).

Paramètre		Description
Kerberos		Spécifie que Kerberos est activé pour les clusters qui utilisent cette configuration de sécurité. Si un cluster utilise cette configuration de sécurité, les paramètres Kerberos doivent également être spécifiés pour celui-ci, sinon, une erreur se produit.
Fournisseur	KDC dédié au cluster	Spécifie que Amazon EMR crée un KDC sur le nœud principal de tout cluster qui utilise cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administrateur KDC au moment de créer le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaires. Créez ces clusters en utilisant une configuration de sécurité différente, spécifiez un KDC externe et utilisez le nom de domaine et le mot de passe administrateur KDC que vous avez spécifiés pour le KDC dédié au cluster.
	KDC externe	Disponible uniquement avec Amazon EMR 5.20.0 et versions ultérieures. Indique que les clusters qui utilisent cette configuration de sécurité authentifient les mandataires Kerberos à l'aide d'un serveur KDC externe au cluster. Aucun KDC n'est créé sur le cluster. Lorsque

Paramètre		Description
		vous créez un cluster, vous spécifiez le nom de domaine et le mot de passe administrateur KDC pour le KDC externe.
Durée de vie du ticket		Facultatif. Spécifie la période pour laquelle un ticket Kerberos émis par le KDC est valide sur les clusters qui utilisent cette configuration de sécurité. La durée de vie des tickets est limitée pour des raisons de sécurité. Les applications et services de cluster renouvellent automatiquement les tickets après leur expiration. Les utilisateurs qui se connectent au cluster via SSH à l'aide d'informations d'identification Kerberos doivent exécuter `kinit` à partir de la ligne de commande du nœud principal pour renouveler un ticket après son expiration.
Relation d'approbation inter-domaines		Spécifie une relation d'approbation inter-domaines entre un KDC dédié au cluster sur les clusters qui utilisent cette configuration de sécurité et un KDC situé dans un autre domaine Kerberos. Les mandataires (généralement des utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est nécessaire. Pour plus d'informations, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277).
Propriétés de la relation d'approbation inter-domaines	Domaine	Spécifie le nom de domaine Kerberos de l'autre domaine inclus dans la relation d'approbation. Par convention, les noms de domaine Kerberos sont identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.
	Domaine	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.
	Serveur d'administration	Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:749`).

Paramètre			Description
	Serveur KDC		Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur KDC de l'autre domaine inclus dans la relation d'approbation. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:88`).
KDC externe			Indique que le KDC externe de cluster est utilisé par le cluster.
Propriétés du KDC externe	Serveur d'administration		Indique le nom de domaine complet (FQDN) ou l'adresse IP du serveur d'administration externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais communiquent sur différents ports. Si aucun port n'est spécifié, le port 749 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:749`).
	Serveur KDC		Indique le nom de domaine complet (FQDN) du serveur KDC externe. Le serveur d'administration et le serveur KDC s'exécutent généralement sur le même poste avec le même FQDN, mais utilisent des ports différents. Si aucun port n'est spécifié, le port 88 est utilisé (port Kerberos par défaut). Le cas échéant, vous pouvez spécifier le port (par exemple, `domain.example.com:88`).
	Intégration d'Active Directory		Indique que l'authentification du mandataire Kerberos est intégrée à un domaine Microsoft Active Directory.
	Propriétés d'intégration d'Active Directory	Active Directory realm (Domaine (realm) Active Directory)	Indique le nom du domaine (realm) Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine classique, sauf qu'ils comportent uniquement des majuscules.
		Active Directory domain (Domaine Active Directory)	Indique le nom de domaine Active Directory.

Paramètre			Description
		Active Directory server (Serveur Active Directory)	Indique le nom de domaine complet (FQDN) du contrôleur de domaine Microsoft Active Directory.

Kerberos Settings for Clusters

Vous pouvez spécifier des paramètres Kerberos à utiliser lorsque vous créez un cluster à l'aide de la console Amazon EMR, de l'AWS CLI ou de l'API EMR.

Utilisez les références suivantes pour comprendre les paramètres de configuration de sécurité de cluster disponibles pour l'architecture Kerberos que vous choisissez. Les paramètres de console Amazon EMR sont affichés. Pour les options d'interface de ligne de commande correspondantes, consultez Configuration Examples (p. 270).

Paramètre :	Description :
Domaine	Nom du domaine Kerberos du cluster. La convention Kerberos consiste à définir un nom identique à celui du domaine, mais en majuscules. Par exemple, pour le domaine `ec2.internal`, on utilise `EC2.INTERNAL` comme nom de domaine.
Mot de passe administrateur du KDC	Le mot de passe utilisé dans le cluster pour `kadmin` ou `kadmin.local`. Il s'agit d'interfaces de ligne de commande vers le système d'administration Kerberos V5, qui maintient les principes de Kerberos, les politiques de mot de passe et les onglets de clavier pour le cluster.
Mot de passe du principal de l'approbation inter-domaines (facultatif)	Obligatoire en cas d'établissement d'une approbation inter-domaines. Mot de passe du principal de l'approbation inter-domaines, qui doit être identique dans tous les domaines. Utilisez un mot de passe fort.
Utilisateur de jonction du domaine Active Directory (facultatif)	Obligatoire lors de l'utilisation d'Active Directory dans une approbation inter-domaines. Il s'agit du nom de connexion d'utilisateur d'un compte Active Directory avec l'autorisation d'ajouter des ordinateurs au domaine. Amazon EMR utilise cette identité pour joindre le cluster au domaine. Pour plus d'informations, consultez the section called "Step 3: Add User Accounts to the Domain for the EMR Cluster" (p. 279),
Mot de passe de jonction du domaine Active Directory (facultatif)	Le mot de passe de l'utilisateur de jonction de domaine Active Directory. Pour plus d'informations, consultez the section called "Step 3: Add User Accounts to the Domain for the EMR Cluster" (p. 279),

Configuration Examples

Les exemples suivants montrent les configurations de sécurité et les configurations de cluster pour des scénarios courants. Les commandes AWS CLI sont affichées par souci de concision.

Local KDC

Les commandes suivantes créent un cluster avec un KDC dédié au cluster qui s'exécute sur le nœud principal. Une configuration supplémentaire du cluster est requise. Pour plus d'informations, consultez Configuring a Cluster for Kerberos-Authenticated HDFS Users and SSH Connections (p. 271),

Create Security Configuration

```
aws emr create-security-configuration --name LocalKDCSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc",\
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24 }}}}'
```

Create Cluster

```
aws emr create-cluster --release-label emr-5.30.1 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive --ec2-attributes
 InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole \
--security-configuration LocalKDCSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyPassword
```

Cluster-Dedicated KDC with Active Directory Cross-Realm Trust

Les commandes suivantes créent un cluster avec un KDC dédié au cluster qui s'exécute sur le nœud principal avec une approbation inter-domaines à un domaine Active Directory. Une configuration supplémentaire sur le cluster et dans Active Directory est requise. Pour plus d'informations, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277),

Create Security Configuration

```
aws emr create-security-configuration --name LocalKDCWithADTrustSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc", \
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24, \
"CrossRealmTrustConfiguration": {"Realm":"AD.DOMAIN.COM", \
"Domain":"ad.domain.com", "AdminServer":"ad.domain.com", \
"KdcServer":"ad.domain.com"}}}}}'
```

Create Cluster

```
aws emr create-cluster --release-label emr-5.30.1 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration KDCWithADTrustSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyClusterKDCAdminPassword,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPassword
```

External KDC on a Different Cluster

Les commandes suivantes créent un cluster qui fait référence à un KDC dédié au cluster sur le nœud principal d'un cluster différent pour authentifier les mandataires. Une configuration supplémentaire du

cluster est requise. Pour plus d'informations, consultez Configuring a Cluster for Kerberos-Authenticated HDFS Users and SSH Connections (p. 271),

Create Security Configuration

```
aws emr create-security-configuration --name ExtKDCOnDifferentCluster \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSOfKDCMaster:749", \
"KdcServer": "MasterDNSOfKDCMaster:88"}}}}'
```

Create Cluster

```
aws emr create-cluster --release-label emr-5.30.1 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCOnDifferentCluster \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword
```

External Cluster KDC with Active Directory Cross-Realm Trust

Les commandes suivantes créent un cluster sans KDC. Le cluster fait référence à un KDC dédié au cluster s'exécutant sur le nœud principal d'un cluster différent pour authentifier les mandataires. Ce KDC dispose d'une approbation inter-domaines avec un contrôleur de domaine Active Directory. Une configuration supplémentaire est requise sur le nœud principal avec le KDC. Pour plus d'informations, consultez Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain (p. 277),

Create Security Configuration

```
aws emr create-security-configuration --name ExtKDCWithADIntegration \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSofClusterKDC:749", \
"KdcServer": "MasterDNSofClusterKDC.com:88", \
"AdIntegrationConfiguration": {"AdRealm":"AD.DOMAIN.COM", \
"AdDomain":"ad.domain.com"}}}}}'
```

Create Cluster

```
aws emr create-cluster --release-label emr-5.30.1 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCWithADIntegration \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword,\
ADDomainJoinUser=MyPrivilegedADUserName,ADDomainJoinPassword=PasswordForADDomainJoinUser
```

Configuring a Cluster for Kerberos-Authenticated HDFS Users and SSH Connections

Amazon EMR crée des clients d'utilisateur authentifiés via Kerberos pour les applications qui s'exécutent sur le cluster—par exemple, l'utilisateur `hadoop`, l'utilisateur `spark` et d'autres encore. Vous pouvez également ajouter des utilisateurs qui sont authentifiés pour les processus de cluster en utilisant Kerberos. Les utilisateurs authentifiés peuvent alors se connecter au cluster avec leurs informations d'identification

Kerberos et utiliser des applications. Les configurations suivantes sont requises pour qu'un utilisateur puisse s'authentifier auprès du cluster :

- A Linux user account matching the Kerberos principal in the KDC must exist on the cluster. Amazon EMR does this automatically in architectures that integrate with Active Directory.
- You must create an HDFS user directory on the master node for each user, and give the user permissions to the directory.
- You must configure the SSH service so that GSSAPI is enabled on the master node. In addition, users must have an SSH client with GSSAPI enabled.

Adding Linux Users and Kerberos Principals to the Master Node

Si vous n'utilisez pas Active Directory, vous devez créer des comptes Linux sur le nœud principal du cluster et ajouter des mandataires au KDC pour ces utilisateurs Linux. Cela comprend un mandataire dans le KDC pour le nœud principal. En plus des mandataires d'utilisateurs, le KDC qui s'exécute sur le nœud principal nécessite un mandataire pour l'hôte local.

Lorsque votre architecture inclut l'intégration d'Active Directory, les utilisateurs et mandataires Linux sur le KDC local sont créés automatiquement, le cas échéant. Vous pouvez ignorer cette étape. Pour de plus amples informations, veuillez consulter Cross-Realm Trust (p. 258) et External KDC—Cluster KDC on a Different Cluster with Active Directory Cross-Realm Trust (p. 263).

> **Important**
>
> Le KDC (ainsi que la base de données de principaux) est perdue lorsque que le nœud principal est arrêté, car ce dernier utilise un magasin éphémère. Si vous créez des utilisateurs pour des connexions SSH, nous vous recommandons d'établir une relation d'approbation inter-domaines avec un KDC externe configuré pour la haute disponibilité. Sinon, si vous créez des utilisateurs pour des connexions SSH à l'aide de comptes d'utilisateur Linux, automatisez le processus de création de compte à l'aide d'actions et de scripts d'amorçage pour pouvoir répéter ce processus lorsque vous créez un nouveau cluster.

La soumission d'une étape au cluster après l'avoir créée ou lorsque vous créez le cluster est la solution la plus simple pour ajouter des utilisateurs et des mandataires KDC. Sinon, vous pouvez vous connecter au nœud principal à l'aide d'une paire de clés EC2 en tant qu'utilisateur `hadoop` par défaut pour exécuter les commandes. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368),

L'exemple suivant envoie un script bash `configureCluster.sh` à un cluster qui existe déjà, en faisant référence à son ID de cluster. Le script est enregistré dans Amazon S3.

```
aws emr add-steps --cluster-id j-01234567 \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://myregion.elasticmapreduce/libs/script-runner/script-runner.jar,\
Args=["s3://mybucket/configureCluster.sh"]
```

L'exemple suivant illustre le contenu du script `configureCluster.sh`. Le script gère également la création des annuaires d'utilisateurs HDFS et l'activation de GSSAPI pour SSH, qui sont abordés dans les sections suivantes.

```
#!/bin/bash
#Add a principal to the KDC for the master node, using the master node's returned host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([lijuan]=pwd1 [marymajor]=pwd2 [richardroe]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
```

```
     name=${i}
     password=${arr[${i}]}

     # Create a principal for each user in the master node and require a new password on
 first logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add hdfs directory for each user
     hdfs dfs -mkdir /user/$name

     #Change owner of each user's hdfs directory to that user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo /etc/init.d/sshd restart
```

Adding User HDFS Directories

Pour autoriser vos utilisateurs à se connecter au cluster afin d'exécuter des travaux Hadoop, vous devez ajouter des annuaires d'utilisateurs HDFS pour leurs comptes utilisateur Linux et accorder à chaque utilisateur la propriété de son annuaire.

La soumission d'une étape au cluster après l'avoir créée ou lorsque vous créez le cluster est la solution la plus simple pour créer des annuaires HDFS. Sinon, vous pourriez vous connecter au nœud principal à l'aide d'une paire de clés EC2 en tant qu'utilisateur `hadoop` par défaut pour exécuter les commandes. Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368),

L'exemple suivant envoie un script bash `AddHDFSUsers.sh` à un cluster qui existe déjà, en faisant référence à son ID de cluster. Le script est enregistré dans Amazon S3.

```
aws emr add-steps --cluster-id ClusterID \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://MyRegion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
MyBucketPath/AddHDFSUsers.sh"]
```

L'exemple suivant illustre le contenu du script `AddHDFSUsers.sh`.

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD, or Linux users created manually on the cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

Enabling GSSAPI for SSH

Afin de permettre aux utilisateurs authentifiés par Kerberos de se connecter au nœud principal à l'aide de SSH, le service SSH doit disposer de l'authentification GSSAPI activée. Pour activer la GSSAPI, exécutez

les commandes suivantes à partir de la ligne de commande du nœud principal ou utilisez une étape pour l'exécuter en tant que script. Une fois que vous avez reconfiguré SSH, vous devez redémarrer le service.

```
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo /etc/init.d/sshd restart
```

Using SSH to Connect to Kerberized Clusters

Cette section illustre les étapes pour qu'un utilisateur authentifié par Kerberos se connecte au nœud principal d'un cluster EMR.

Chaque ordinateur qui est utilisé pour une connexion SSH doit avoir le client SSH et les applications client Kerberos installés. Il est probable que les ordinateurs Linux comportent ces éléments par défaut. Par exemple, OpenSSH est installé sur la plupart des systèmes d'exploitation Linux, Unix et macOS. Vous pouvez vérifier un client SSH en tapant ssh dans la ligne de commande. Si votre ordinateur ne reconnaît pas la commande, installez un client SSH pour vous connecter au nœud principal. Le projet OpenSSH offre une implémentation gratuite de la suite entière des outils SSH. Pour plus d'informations, consultez le site Web OpenSSH. Les utilisateurs Windows peuvent utiliser des applications telles que PuTTY en tant que client SSH.

Pour plus d'informations sur vos connexions SSH, consultez Connexion au cluster (p. 367).

SSH utilise GSSAPI pour authentifier les clients Kerberos et vous devez activer l'authentification GSSAPI pour le service SSH sur le nœud principal du cluster. Pour plus d'informations, consultez Enabling GSSAPI for SSH (p. 273), Les clients SSH doivent également utiliser GSSAPI.

Dans les exemples suivants, pour *`MasterPublicDNS`* utiliser la valeur qui apparaît pour DNS public principal sur le Résumé onglet du volet Détails du cluster—par exemple, *`ec2-11-222-33-44.compute-1.amazonaws.com`*.

Prerequisite for krb5.conf (Non Active Directory)

Lorsque vous utilisez une configuration sans l'intégration d'Active Directory, en plus du client SSH et des applications clientes Kerberos, chaque ordinateur client doit avoir une copie du fichier `/etc/krb5.conf` correspondant au fichier `/etc/krb5.conf` sur le nœud principal du cluster.

Pour copier le fichier krb5.conf

1. Utilisez SSH pour vous connecter au nœud maître à l'aide d'une paire de clés EC2 et de la valeur par défaut `hadoop` utilisateur—par exemple, `hadoop@`*`MasterPublicDNS`*. Pour des instructions détaillées, voir Connexion au cluster (p. 367).
2. Dans le nœud principal, copiez le contenu du fichier `/etc/krb5.conf`. Pour plus d'informations, consultez Connexion au cluster (p. 367),
3. Sur chaque ordinateur client qui sera utilisé pour se connecter au cluster, créez un fichier `/etc/krb5.conf` identique à partir de la copie que vous avez créée à l'étape précédente.

Using Kinit and SSH

Chaque fois qu'un utilisateur se connecte à partir d'un ordinateur client à l'aide des informations d'identification Kerberos, l'utilisateur doit d'abord renouveler un ticket Kerberos pour leurs utilisateurs sur l'ordinateur client. En outre, le client SSH doit être configuré pour utiliser l'authentification GSSAPI.

Utilisation de SSH pour se connecter aux clusters EMR Kerberos

1. Utilisez `kinit` pour renouveler vos tickets Kerberos, comme illustré dans l'exemple suivant

```
kinit user1
```

2. Utilisez un client `ssh` en même temps que le mandataire que vous avez créé dans le KDC dédié au cluster ou dans le nom d'utilisateur Active Directory. Assurez-vous que l'authentification GSSAPI est activée, telle qu'illustrée dans les exemples suivants.

   Example: Linux users

   L'option `-K` spécifie l'authentification GSSAPI.

```
ssh -K user1@MasterPublicDNS
```

   Example: Windows users (PuTTY)

   Assurez-vous que l'authentification GSSAPI est activée pour la session, telle qu'illustrée :

Tutorial: Configure a Cluster-Dedicated KDC

Cette rubrique vous guide dans la création d'un cluster avec un cluster dédié key distribution center (KDC), ajoutant manuellement des comptes utilisateur Linux à tous les nœuds de cluster, en ajoutant des principes Kerberos au KDC sur le noeud maître, et en s'assurant que les ordinateurs client ont un client Kerberos installé.

Pour plus d'informations sur le support d'Amazon EMR pour Kerberos et KDC, ainsi que des liens vers la documentation MIT Kerberos, voir Use Kerberos Authentication (p. 255).

Step 1: Create the Kerberized Cluster

1. Créez une configuration de sécurité qui active Kerberos. L'exemple suivant illustre une commande `create-security-configuration` à partir de l'AWS CLI qui spécifie la configuration de sécurité sous forme de structure JSON en ligne. Vous pouvez également référencer un fichier enregistré localement.

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{"AuthenticationConfiguration": {"KerberosConfiguration":
{"Provider": "ClusterDedicatedKdc", "ClusterDedicatedKdcConfiguration":
 {"TicketLifetimeInHours": 24}}}}}'
```

2. Créez un cluster qui fait référence à la configuration de sécurité, établit les attributs Kerberos du cluster et ajoute des comptes Linux à l'aide d'une action d'amorçage. L'exemple suivant illustre l'utilisation d'une commande `create-cluster` à partir de l'AWS CLI. La commande renvoie à la configuration de sécurité que vous avez créée ci-dessus, `MyKerberosConfig`. Il fait également référence à un script simple, `createlinuxusers.sh`, comme action d'amorçage, que vous créez et téléchargez pour Amazon S3 avant de créer le cluster.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.30.1 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair \
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd \
--bootstrap-actions Path=s3://mybucket/createlinuxusers.sh
```

L'exemple suivant illustre le contenu du script `createlinuxusers.sh` qui ajoute user1, user2 et user3 à chaque nœud du cluster. Dans l'étape suivante, vous ajoutez ces utilisateurs en tant que principaux KDC.

```
#!/bin/bash
sudo adduser user1
sudo adduser user2
sudo adduser user3
```

Step 2: Add Principals to the KDC, Create HDFS User Directories, and Configure SSH

Le KDC qui s'exécute sur le nœud principal requiert que vous ajoutiez un principal pour l'hôte local et pour chaque utilisateur que vous créez sur le cluster. Vous pouvez également créer des annuaires HDFS pour chaque utilisateur qui a besoin de se connecter au cluster et d'exécuter des travaux Hadoop. De même, configurez le service SSH pour activer l'authentification GSSAPI, qui est obligatoire pour Kerberos. Une fois que vous avez activé GSSAPI, redémarrez le service SSH.

La manière la plus simple d'effectuer ces tâches est d'envoyer une étape au cluster. L'exemple suivant envoie un script bash `configurekdc.sh` au cluster que vous avez créé dans l'étape précédente, en faisant référence à son ID de cluster. Le script est enregistré dans Amazon S3. Sinon, vous pouvez vous connecter au nœud principal à l'aide d'une paire de clés EC2 pour exécuter les commandes ou envoyer l'étape lors de la création du cluster.

```
aws emr add-steps --cluster-id j-01234567 --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myregion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://mybucket/
configurekdc.sh"]
```

L'exemple suivant illustre le contenu du script `configurekdc.sh`.

```
#!/bin/bash
#Add a principal to the KDC for the master node, using the master node's returned host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([user1]=pwd1 [user2]=pwd2 [user3]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
     name=${i}
     password=${arr[${i}]}

     # Create principal for sshuser in the master node and require a new password on first
 logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add user hdfs directory
     hdfs dfs -mkdir /user/$name

     #Change owner of user's hdfs directory to user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo /etc/init.d/sshd restart
```

Les utilisateurs que vous avez ajoutés doivent maintenant être en mesure de se connecter au cluster à l'aide de SSH. Pour plus d'informations, consultez Using SSH to Connect to Kerberized Clusters (p. 274),

Tutorial: Configure a Cross-Realm Trust with an Active Directory Domain

Lorsque vous configurez une approbation inter-domaines, vous autorisez des principaux (généralement des utilisateurs) provenant d'un autre domaine Kerberos à s'authentifier auprès de composants d'application sur le cluster EMR. Le cluster dédié key distribution center (KDC) établit une relation de confiance avec un autre KDC en utilisant cross-realm principal qui existe dans les deux kdcs. Le nom et le mot de passe du principal correspondent exactement.

Une approbation inter-domaines nécessite que les KDC puissent s'atteindre mutuellement via le réseau et résoudre leurs noms de domaine mutuels. Les étapes permettant d'établir une relation d'approbation inter-domaines avec un contrôleur de domaine Microsoft AD s'exécutant en tant qu'instance EC2 sont fournies ci-dessous, ainsi qu'un exemple de configuration de réseau qui fournit la connectivité et la résolution du nom de domaine requises. Toute configuration de réseau qui autorise le trafic réseau requis entre les KDC est acceptable.

Le cas échéant, une fois l'approbation inter-domaines établie avec Active Directory à l'aide d'un KDC sur un cluster, vous pouvez créer un autre cluster à l'aide d'une autre configuration de sécurité pour référencer le KDC sur le premier cluster comme KDC externe. Pour un exemple de configuration de sécurité et d'installation de cluster, consultez External Cluster KDC with Active Directory Cross-Realm Trust (p. 271).

Pour plus d'informations sur le support d'Amazon EMR pour Kerberos et KDC, ainsi que des liens vers la documentation MIT Kerberos, voir Use Kerberos Authentication (p. 255).

> **Important**
>
> Amazon EMR ne prend pas en charge les approbations inter-domaines avec AWS Directory Service for Microsoft Active Directory (Enterprise Edition).

Step 1: Set Up the VPC and Subnet (p. 278)

Step 2: Launch and Install the Active Directory Domain Controller (p. 279)

Step 3: Add User Accounts to the Domain for the EMR Cluster (p. 279)

Step 4: Configure an Incoming Trust on the Active Directory Domain Controller (p. 280)

Step 5: Use a DHCP Option Set to Specify the Active Directory Domain Controller as a VPC DNS Server (p. 280)

Step 6: Launch a Kerberized EMR Cluster (p. 280)

Step 7: Create HDFS Users and Set Permissions on the Cluster for Active Directory User Accounts (p. 281)

Step 1: Set Up the VPC and Subnet

Les étapes suivantes montrent la création d'un VPC et d'un sous-réseau afin que le KDC dédié au cluster puisse atteindre le contrôleur de domaine Active Directory et résoudre son nom de domaine. Au cours de ces étapes, la résolution du nom de domaine est fournie en faisant référence au contrôleur de domaine Active Directory en tant que serveur de noms de domaine dans le jeu d'options DHCP. Pour plus d'informations, consultez Step 5: Use a DHCP Option Set to Specify the Active Directory Domain Controller as a VPC DNS Server (p. 280),

Le KDC et le contrôleur de domaine Active Directory doivent être en mesure de résoudre leurs noms de domaine respectifs. Ceci permet à Amazon EMR d'associer des ordinateurs au domaine et de configurer automatiquement les comptes d'utilisateur Linux et les paramètres SSH correspondants sur les instances de cluster.

Si Amazon EMR ne peut pas résoudre le nom de domaine, vous pouvez référencer l'approbation à l'aide de l'adresse IP du contrôleur de domaine Active Directory. Cependant, vous devez ajouter manuellement les comptes d'utilisateurs Linux, ajouter les principaux correspondants au KDC dédié au cluster et configurer SSH.

Pour configurer le VPC et le sous-réseau

1. Créez un Amazon VPC avec un seul sous-réseau public. Pour plus d'informations, voir Étape 1 : Créer le VPC dans le Amazon VPC Guide de mise en route.

   > **Important**
   >
   > Lorsque vous utilisez un contrôleur de domaine Microsoft Active Directory, choisissez un bloc d'adresse CIDR pour le cluster EMR de sorte que toutes les adresses IPv4 aient moins de neuf caractères (par exemple, 10.0.0.0/16). Cela provient du fait que les noms DNS des ordinateurs en cluster sont utilisés lorsque les ordinateurs rejoignent l'annuaire Active Directory. AWS affecte les noms d'hôte DNS en fonction de l'adresse IPv4 de telle manière que les adresses IP plus longues peuvent entraîner des noms DNS de plus de 15 caractères. Active Directory a une limite de 15 caractères pour l'enregistrement des noms d'ordinateurs joints et tronque les noms plus longs, ce qui peut provoquer des erreurs imprévisibles.

2. Supprimez le jeu d'options DHCP par défaut attribué au VPC. Pour plus d'informations, consultez Modification d'un VPC pour qu'il n'utilise pas d'options DHCP. Par la suite, vous ajoutez un nouveau jeu d'options qui spécifie le contrôleur de domaine Active Directory en tant que serveur DNS.
3. Vérifiez que la prise en charge de DNS est activée pour le VPC, c'est-à-dire que les noms d'hôte DNS et la résolution DNS sont activés. Ils sont activés par défaut. Pour plus d'informations, consultez Mise à jour de la prise en charge de DNS de votre VPC.
4. Vérifiez que votre VPC dispose d'une passerelle Internet attachée (c'est le cas par défaut). Pour plus d'informations, consultez Création et attachement d'une passerelle Internet.

   Note

   Une passerelle Internet est utilisée dans cet exemple, car vous établissez un nouveau contrôleur de domaine pour le VPC. Il peut cependant arriver qu'aucune passerelle Internet ne soit requise pour votre application. La seule exigence est que le KDC dédié au cluster puisse accéder au contrôleur de domaine Active Directory.
5. Créez une table de routage personnalisée, ajoutez une route qui mène à la passerelle Internet, puis attachez-la à votre sous-réseau. Pour plus d'informations, consultez Création d'une table de routage personnalisée.
6. Lorsque vous lancez l'instance EC2 pour le contrôleur de domaine, elle doit avoir une adresse IPv4 publique statique pour que vous puissiez vous y connecter à l'aide de RDP. La manière la plus simple de procéder est de configurer votre sous-réseau afin qu'il attribue automatiquement des adresses IPv4 publiques. Ce n'est pas le paramètre par défaut lorsqu'un sous-réseau est créé. Pour plus d'informations, consultez Modification de l'attribut d'adressage IPv4 public de votre sous-réseau. Vous avez aussi la possibilité d'attribuer l'adresse lorsque vous lancez l'instance. Pour plus d'informations, consultez Attribution d'une adresse IPv4 publique lors du lancement d'une instance.
7. Lorsque vous avez fini, notez les ID de votre VPC et du sous-réseau. Vous les utiliserez ultérieurement lorsque vous lancerez le contrôleur de domaine Active Directory et le cluster.

Step 2: Launch and Install the Active Directory Domain Controller

1. Lancez une instance EC2 à partir de l'AMI de base de Microsoft Windows Server 2016. Nous vous recommandons le type d'instance m4.xlarge ou plus. Pour plus d'informations, consultez Lancement d'une instance AWS Marketplace dans le Amazon EC2 Guide de l'utilisateur pour les instances Windows.
2. Notez l'ID de groupe du groupe de sécurité associé à l'instance EC2. Vous en avez besoin pour Step 6: Launch a Kerberized EMR Cluster (p. 280). Nous utilisons `sg-012xrlmdomain345`Vous pouvez aussi spécifier différents groupes de sécurité pour le cluster EMR et cette instance qui autorise le trafic entre eux. Pour plus d'informations, consultez Groupes de sécurité Amazon EC2 pour les instances Linux dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
3. Connectez-vous à l'instance EC2 à l'aide de RDP. Pour plus d'informations, consultez Connexion à votre instance Windows dans le Amazon EC2 Guide de l'utilisateur pour les instances Windows.
4. Démarrez le Gestionnaire de serveurs pour installer et configurer le rôle des services de domaine Active Directory sur le serveur. Configurez le serveur comme contrôleur de domaine et attribuez un nom de domaine (l'exemple que nous utilisons ici est `ad.domain.com`). Notez le nom de domaine, car vous en aurez besoin plus tard lorsque vous créerez la configuration de sécurité et le cluster EMR. Si vous configurez Active Directory pour la première fois, vous pouvez suivre les instructions indiquées dans Comment configurer Active Directory (AD) dans Windows Server 2016.

   L'instance redémarre une fois que vous avez terminé.

Step 3: Add User Accounts to the Domain for the EMR Cluster

RDP sur le contrôleur de domaine Active Directory pour créer des comptes d'utilisateur dans les utilisateurs et ordinateurs Active Directory pour chaque utilisateur de cluster. Pour plus d'informations, consultez Créer

un compte d'utilisateur dans Utilisateurs et ordinateurs Active Directory. Notez le nom de connexion de chaque utilisateur. Vous en aurez besoin ultérieurement lorsque vous configurez le cluster.

En outre, créez un compte utilisateur avec des privilèges suffisants pour joindre des ordinateurs au domaine. Vous spécifiez ce compte lorsque vous créez un cluster. Amazon EMR l'utilise pour joindre les instances de cluster au domaine. Vous spécifiez ce compte et son mot de passe Step 6: Launch a Kerberized EMR Cluster (p. 280). Pour déléguer des privilèges de jointure d'ordinateur au compte d'utilisateur, nous vous recommandons de créer un groupe avec des privilèges de jointure, puis d'affecter l'utilisateur au groupe. Pour plus d'informations, consultez Délégation des privilèges de jonction de répertoire dans le AWS Directory Service Administration Guide.

Step 4: Configure an Incoming Trust on the Active Directory Domain Controller

L'exemple des commandes ci-dessous permet de créer une relation d'approbation dans Active Directory, qui est une approbation de domaine unidirectionnelle, entrante et non transitive avec le KDC dédié au cluster. L'exemple que nous utilisons pour le domaine du cluster est *`EC2.INTERNAL`*. Remplacer le *`KDC-FQDN`* avec le DNS public nom indiqué pour le Amazon EMR noeud principal hébergeant le KDC. Le paramètre `passwordt` spécifie le mot de passe du mandataire inter-domaines, que vous spécifiez en même temps que le domaine du cluster lorsque vous créez un cluster. Le nom de domaine est dérivé du nom de domaine par défaut dans `us-east-1` pour le cluster. Le `Domain` est le domaine Active Directory dans lequel vous créez la stratégie d'approbation qui est en minuscules par convention. L'exemple utilise *`ad.domain.com`*

Ouvrez l'invite de commande Windows avec des privilèges d'administrateur et entrez les commandes suivantes pour créer la relation d'approbation sur le contrôleur de domaine Active Directory :

```
C:\Users\Administrator> ksetup /addkdc EC2.INTERNAL KDC-FQDN
C:\Users\Administrator> netdom trust EC2.INTERNAL /Domain:ad.domain.com /add /realm /
passwordt:MyVeryStrongPassword
C:\Users\Administrator> ksetup /SetEncTypeAttr EC2.INTERNAL AES256-CTS-HMAC-SHA1-96
```

Step 5: Use a DHCP Option Set to Specify the Active Directory Domain Controller as a VPC DNS Server

Maintenant que le contrôleur de domaine Active Directory est configuré, vous devez configurer le VPC afin de l'utiliser comme serveur de nom de domaine pour la résolution des noms au sein de votre VPC. Pour ce faire, attachez un jeu d'options DHCP. Indiquez le nom de domaine comme nom de domaine de votre cluster (par exemple, `ec2.internal` si votre cluster se trouve dans la région us-est-1 ou *`region`*`.compute.internal` pour les autres régions). Pour Serveurs de noms de domaine, vous devez spécifier l'adresse IP du contrôleur de domaine Active Directory (qui doit être accessible du cluster) comme première entrée, suivie de amazondenddns (par exemple, ***`xx.xx.xx.xx`***, hypnotisés). Pour plus d'informations, consultez Modification des jeux d'options DHCP.

Step 6: Launch a Kerberized EMR Cluster

1. Dans Amazon EMR, créez une configuration de sécurité qui spécifie le contrôleur de domaine Active Directory que vous avez créé dans les étapes précédentes. Un exemple de commande est présenté ci-dessous. Remplacez le domaine, *`ad.domain.com`*, par le nom du domaine que vous avez spécifié dans Step 2: Launch and Install the Active Directory Domain Controller (p. 279).

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
```

```
            "Domain": "ad.domain.com",
            "AdminServer": "ad.domain.com",
            "KdcServer": "ad.domain.com"
          }
        }
      }
    }
}'
```

2. Créez le cluster avec les attributs suivants :
   - Use the `--security-configuration` option to specify the security configuration that you created. We use *`MyKerberosConfig`* in the example.
   - Use the `SubnetId` property of the `--ec2-attributes option` to specify the subnet that you created in Step 1: Set Up the VPC and Subnet (p. 278). We use *`step1-subnet`* in the example.
   - Use the `AdditionalMasterSecurityGroups` and `AdditionalSlaveSecurityGroups` of the `--ec2-attributes` option to specify that the security group associated with the AD Domain Controller from Step 2: Launch and Install the Active Directory Domain Controller (p. 279) is associated with the cluster master node as well as core and task nodes. We use *`sg-012xrlmdomain345`* in the example.

   Utilisez `--kerberos-attributes` pour spécifier les attributs Kerberos suivants spécifiques au cluster :
   - The realm for the cluster that you specified when you set up the Active Directory domain controller.
   - The cross-realm trust principal password that you specified as `passwordt` in Step 4: Configure an Incoming Trust on the Active Directory Domain Controller (p. 280).
   - A `KdcAdminPassword`, which you can use to administer the cluster-dedicated KDC.
   - The user logon name and password of the Active Directory account with computer join privileges that you created in Step 3: Add User Accounts to the Domain for the EMR Cluster (p. 279).

   L'exemple suivant lance un cluster activé pour Kerberos.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.10.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair,\
SubnetId=step1-subnet, AdditionalMasterSecurityGroups=sg-012xrlmdomain345,
AdditionalSlaveSecurityGroups=sg-012xrlmdomain345\
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPwd
```

Step 7: Create HDFS Users and Set Permissions on the Cluster for Active Directory User Accounts

Lors de la mise en place d'une relation d'approbation avec Active Directory, Amazon EMR crée des utilisateurs Linux sur le cluster pour chaque compte d'utilisateur Active Directory. Par exemple, le nom de connexion utilisateur `LiJuan` dans Active Directory possède un compte utilisateur Linux `lijuan`. Les noms d'utilisateur Active Directory peuvent contenir des lettres majuscules, mais Linux n'honore pas le boîtier Active Directory.

Pour autoriser vos utilisateurs à se connecter au cluster afin d'exécuter des travaux Hadoop, vous devez ajouter des annuaires d'utilisateurs HDFS pour leurs comptes utilisateur Linux et accorder à chaque

utilisateur la propriété de son annuaire. Pour ce faire, nous vous recommandons d'exécuter un script enregistré dans Amazon S3 sous la forme d'une étape de cluster. Sinon, vous pouvez exécuter les commandes dans le script ci-dessous à partir de l'interface de ligne de commande sur le nœud principal. Utilisez la paire de clés EC2 que vous avez spécifiée lors de la création du cluster pour vous connecter au nœud principal via SSH en tant qu'utilisateur Hadoop. Pour plus d'informations, consultez Use an Amazon EC2 Key Pair for SSH Credentials (p. 255),

Exécutez la commande suivante pour ajouter une étape au cluster qui exécute un script, *AddHDFSUsers.sh*.

```
aws emr add-steps --cluster-id ClusterID \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://MyRegion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
MyBucketPath/AddHDFSUsers.sh"]
```

Le contenu du fichier *AddHDFSUsers.sh* est le suivant :

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD or Linux users and KDC principals created
 manually on the cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

Active Directory Groups Mapped to Hadoop Groups

Amazon EMR utilise System Security Services Daemon (SSD) pour mapper des groupes Active Directory aux groupes Hadoop. Pour confirmer des mappages de groupe, après la connexion au nœud maître telle que décrite dans Using SSH to Connect to Kerberized Clusters (p. 274), vous pouvez utiliser la commande `hdfs groups` pour confirmer que les groupes Active Directory auxquels votre compte Active Directory appartient ont été mappés aux groupes Hadoop de l'utilisateur Hadoop correspondant sur le cluster. Vous pouvez également vérifier les mappages de groupe d'autres utilisateurs en spécifiant un ou plusieurs noms d'utilisateurs avec la commande, par exemple `hdfs groups lijuan`. Pour plus d'informations, voir groupes dans le Guide des commandes Apache HDFS.

Intégration Amazon EMR avec AWS Lake Formation (Bêta)

À partir d'Amazon EMR 5.26.0, vous pouvez lancer un cluster qui s'intègre à AWS Lake Formation. Cette fonction est disponible en version bêta publique.

AWS Lake Formation est un service géré qui simplifie la configuration, la sécurisation et la gestion des lacs de données. AWS Lake Formation vous aide à découvrir, cataloguer, nettoyer et sécuriser les données dans un Amazon Simple Storage Service (Amazon S3) lac de données. Pour plus d'informations, consultez ,AWS Lake Formation.

Intégration Amazon EMR avec AWS Lake Formation offre les avantages clés suivants :

- Fournit un accès à des colonnes et des bases de données de niveau colonne à grain fin dans le AWS Glue Catalogue de données.
- Active l'authentification unique fédérée sur les blocs-notes EMR ou Apache Zeppelin à partir de votre système d'identité d'entreprise compatible avec SAML (Security Assertion Markup Language) 2.0.

Cette section fournit un aperçu conceptuel de Amazon EMR intégration avec Lake Formation. Elle fournit également les prérequis et les étapes nécessaires au lancement d'un cluster Amazon EMR intégré à Lake Formation.

Rubriques

- Présentation conceptuelle de l'intégration Amazon EMR à Lake Formation (p. 283)
- Applications et fonctionnalités prises en charge (p. 288)
- Avant de commencer (p. 289)
- Lancez un cluster Amazon EMR avec Lake Formation (p. 296)

Présentation conceptuelle de l'intégration Amazon EMR à Lake Formation

En intégrant Amazon EMR avec AWS Lake Formation, vous activez l'authentification SAML avec vos identifiants d'entreprise et appliquez un contrôle d'accès de colonne à grain fin basé sur les politiques définies dans AWS Lake Formation.

Intégration Amazon EMR et Lake Formation, votre organisation doit satisfaire aux exigences suivantes :

- Gérer vos identités d'entreprise à l'aide d'un fournisseur d'identité basé sue SAML existant, tel qu'Active Directory Federation Services (AD FS). Pour plus d'informations, consultez Fournisseurs tiers pris en charge pour SAML (p. 294).
- Utiliser le catalogue de données AWS Glue comme magasin de métadonnées.
- Utiliser les blocs-notes EMR ou Apache Zeppelin pour accéder aux données gérées par AWS Glue et Lake Formation.
- Définir et gérer les autorisations dans Lake Formation pour accéder aux bases de données, aux tableaux et aux colonnes dans AWS Glue Catalogue de données. Pour plus d'informations, consultez ,AWS Lake Formation.

Avant de lancer un cluster qui s'intègre à Lake Formation, vous devez activer la fédération basée sur SAML 2.0 en configurant vos rôles de fournisseur d'identité (IdP) et des rôles AWS Access and Identity Management (IAM). Vous devez également configurer des contrôles de sécurité appropriés pour votre Amazon EMR groupe. Pour plus d'informations, voir Avant de commencer (p. 289) et Lancez un cluster Amazon EMR avec Lake Formation (p. 296).

Rubriques

- Termes et concepts (p. 283)
- Fonctionnement de l'accès aux données dans Lake Formation (p. 284)
- Amazon EMRComposants d' (p. 285)
- Architecture d'authentification unique compatible avec SAML et contrôle précis des accès (p. 286)

Termes et concepts

Cette section offre une vue d'ensemble des concepts et des termes utilisés dans le contexte de l'intégration d'Amazon EMR à AWS Lake Formation.

Authentication

Processus d'établissement de l'identité d'un utilisateur. En intégrant Amazon EMR et Lake Formation, vos utilisateurs peuvent utiliser leurs identifiants d'entreprise pour se connecter aux carnets DME et Apache Zeppelin.

Authorization, *

Processus de validation des actions qu'un utilisateur donné peut effectuer sur certaines ressources. Lors de l'intégration d'un Amazon EMR groupe avec Lake Formation, l'accès aux bases de données et aux tables est autorisé en utilisant les politiques de formation du lac. Ce processus garantit que les utilisateurs peuvent uniquement interroger et analyser les tables ou colonnes auxquelles ils sont autorisés à accéder.

Fédération

La création d'une relation de confiance entre un fournisseur d'identité externe et AWS Identity and Access Management(IAM). Les utilisateurs peuvent également se connecter à un système d'identité d'entreprise compatible avec SAML (Security Assertion Markup Language) 2.0 tel que les services ADFS (Active Directory Federation Services) de Microsoft. Pour plus d'informations, consultez Fournisseurs tiers pris en charge pour SAML (p. 294). Lorsque vous utilisez SAML 2.0 pour configurer une relation d'approbation entre ces fournisseurs d'identité externes et IAM, l'utilisateur est affecté à un rôle IAM. L'utilisateur reçoit également des informations d'identification temporaires qui permettent à l'utilisateur d'accéder à AWS Lake Formation ressources.

Stratégie d'approbation

Un document dans Json format dans lequel vous définissez qui est autorisé à assumer le rôle. Cette entité de confiance est incluse dans la stratégie comme élément principal du document. Le document est rédigé conformément aux règles du Langue de la politique IAM.

Stratégie d'autorisations

Un document d'autorisations dans Json format dans lequel vous définissez les actions et les ressources auxquelles le rôle a accès à. Le document est rédigé conformément aux règles du Langue de la politique IAM.

Principal()

Entité qui peut accéder aux ressources protégées par des stratégies Lake Formation et exécuter des requêtes dans Amazon EMR. Les mandataires peuvent être des utilisateurs ou des rôles AWS Identity and Access Management (IAM), ou des utilisateurs fédérés identifiés par leur fournisseur d'identité (IdP) SAML.

Fonctionnement de l'accès aux données dans Lake Formation

Lake Formation autorise l'accès aux données en fournissant des informations d'identification temporaires à des services tels qu'.Amazon EMR. Ce processus est connu sous le nom de distributeur d'informations d'identification. Pour plus d'informations, consultez ,AWS Lake Formation.

Lorsque vous exécutez une requête sur des données protégées par Lake Formation politiques de sécurité, Amazon EMR demande des informations d'identification temporaires AWS Lake Formation pour accéder aux données stockées dans Amazon S3.

Voici comment l'accès aux données est accordé :

1. Vous configurez et contrôlez l'accès utilisateur aux ressources en utilisant AWS Lake Formation. Vous pouvez créer les politiques en utilisant un ensemble d'autorisations de délégation et de révocation qui vous sont disponibles dans le Lake Formation Console. Par exemple, vous pouvez accorder l'accès à une base de données ou à une table. Vous pouvez également accorder des autorisations au niveau des colonnes aux utilisateurs. Vous spécifiez les autorisations pour les tables et les colonnes directement dans Lake Formationau lieu de les spécifier pour Amazon S3 les godets et les objets. Pour plus d'informations, voir Lake Formation Autorisations.
2. Lorsqu'un principal tente d'exécuter une requête dans Amazon EMR sur les données de Lake Formation, Amazon EMR demande des informations d'identification temporaires pour l'accès aux données depuis AWS Lake Formation.
3. Lake Formation renvoie des informations d'identification temporaires, ce qui permet l'accès aux données.
4. Amazon EMR envoie la demande de requête pour obtenir des données à partir d'Amazon S3.
5. Amazon EMR filtre et renvoie les résultats en fonction des autorisations utilisateur définies dans Lake Formation.

Amazon EMRComposants d'

Amazon EMR permet un contrôle d'accès précis avec Lake Formation en utilisant les composants suivants :

- Agent proxy -L'agent proxy est basé sur Apache Knox. Il reçoit des demandes authentifiées SAML de la par des utilisateurs et traduit les demandes SAML en informations d'identification temporaires. Il stocke également les informations d'identification temporaires dans l'agent secret. L'agent proxy s'exécute sur le noeud principal comme `knox` utilisateur système et écrit les journaux sur `/var/log/knox` répertoire.
- Agent secret -L'agent secret stocke en toute sécurité les secrets et distribue des secrets à d'autres composants ou applications de DME. Les secrets peuvent inclure des informations d'identification utilisateur temporaires, des clés de chiffrement ou des tickets Kerberos. L'agent secret s'exécute sur chaque noeud du cluster et utilise Lake Formation et AWS Glue des API pour récupérer les informations d'identification temporaires et AWS Glue Métadonnées du catalogue de données. L'agent secret fonctionne comme `emrsecretagent` utilisateur, et écrit les journaux sur `/emr/secretagent/log` répertoire. Le processus s'appuie sur un ensemble spécifique de règles `iptables` pour fonctionner.

Il est important de garantir `iptables` n'est pas désactivé et, si vous personnalisez `iptables` configuration, `nat` les règles de la table doivent être conservées et laissées inchangées.

- Serveur d'enregistrement -Le serveur d'enregistrement reçoit des demandes d'accès aux données. Il autorise ensuite les demandes en fonction des informations d'identification temporaires et des stratégies de contrôle d'accès aux tables distribuées par l'agent secret. Le serveur d'enregistrement lit les données dans Amazon S3 et renvoie les données au niveau des colonnes auxquelles l'utilisateur est autorisé à accéder. Le serveur d'enregistrements s'exécute sur chaque noeud du cluster comme `emr_record_server` l'utilisateur et écrit les journaux sur `/var/log/emr-record-server` répertoire.

  Note

  Spark SQL a été intégré à chacun de ces composants, ce qui permet à Spark SQL de lire et traiter les données protégées par Lake Formation.

Architecture d'authentification unique compatible avec SAML et contrôle précis des accès

Le schéma suivant illustre une architecture d'authentification unique compatible avec SAML et un contrôle précis des accès avec Lake Formation et Amazon EMR.

1. Un utilisateur non authentifié utilise l'agent proxy pour accéder au bloc-notes EMR ou à Zeppelin. L'utilisateur est redirigé vers la page de connexion du fournisseur d'identité (IdP) de votre organisation.
2. Le fournisseur d'identité vérifie l'identité de l'utilisateur dans votre organisation.
3. Il génère ensuite une réponse d'authentification SAML contenant des assertions qui identifient l'utilisateur et incluent des attributs spécifiques à celui-ci.
4. Le navigateur client publie l'assertion SAML sur l'agent proxy.

5. L'agent proxy demande à AWS Lake Formation des informations d'identification de sécurité temporaires spécifiques à l'utilisateur au nom de celui-ci. Les informations d'identification de sécurité temporaires sont renvoyées à l'agent proxy.
6. L'agent proxy stocke les informations d'identification de sécurité temporaires spécifiques à l'utilisateur dans l'agent secret. L'agent secret envoie les informations d'identification utilisateur temporaires aux agents secrets dans les nœuds principaux et de tâches.
7. L'agent proxy active la connexion utilisateur.
8. Lorsque l'utilisateur exécute une tâche Spark à l'aide du bloc-notes EMR ou de Zeppelin, le serveur d'enregistrement appelle l'agent secret pour obtenir des informations d'identification utilisateur temporaires.
9. Le serveur d'enregistrement lit et filtre les données depuis Amazon S3 en fonction des stratégies définies dans Lake Formation.

Ce processus est entièrement transparent du point de vue de l'utilisateur. L'utilisateur démarre sur la page d'authentification de votre organisation et accède au bloc-notes EMR ou à Zeppelin via le navigateur sans avoir à fournir d'informations d'identification AWS.

Applications et fonctionnalités prises en charge

Applications prises en charge

L'intégration entre Amazon EMR et AWS Lake Formation prend en charge les applications suivantes :

- Apache Spark
- Apache Zeppelin
- Amazon EMRBlocs-notes

> **Important**
>
> Les autres applications ne sont actuellement pas prises en charge. Pour garantir la sécurité de votre cluster, n'installez pas d'applications autres que celles de cette liste.

Fonctionnalités prises en charge

Les fonctionnalités Amazon EMR suivantes peuvent être utilisées avec EMR et Lake Formation:

- Chiffrement au repos et en transit
- Authentification Kerberos à l'aide d'un KDC dédié au cluster
- Groupes d'instances, parcs d'instances et instances spot
- Reconfiguration d'applications dans un cluster en cours d'exécution

Les fonctions EMR suivantes ne fonctionnent pas actuellement avec Lake Formation intégration :

- Étapes
- Plusieurs nœuds principaux
- Vue cohérente d'EMRFS
- EMRFS CSE-C et SSE-C utilisant des clés de chiffrement fournies par le client

Limitations

Tenez compte des limitations suivantes lors de l'utilisation Amazon EMR avec AWS Lake Formation:

- Dans Lake Formation les clusters activés, Spark SQL ne peut lire que des données gérées par AWS Glue Catalogue des données et ne pas accéder aux données gérées en dehors de la colle AWS ou de la formation du lac. Les données provenant d'autres sources sont accessibles à l'aide d'opérations SQL autres que Spark si le rôle IAM pour les autres services AWS choisi lors du déploiement du cluster dispose de stratégies en place permettant au cluster d'accéder à ces sources de données.
  - Par exemple, vous pourriez avoir deux Amazon S3 les seaux et Amazon DynamoDB tableau que vous souhaitez que votre tâche Spark soit accessible en plus d'un ensemble de Lake Formation tableaux. Dans ce cas, vous pouvez créer un rôle qui peut accéder aux deux Amazon S3 et les Amazon DynamoDB et utilisez-le pour le `IAM role for other AWS services` lors du lancement de votre cluster.
- La soumission de tâches Spark doit être effectuée via des blocs-notes EMR, Zeppelin ou Livy. Travaux Spark soumis par `spark-submit` ne fonctionnera pas avec Lake Formation pour le moment.
- Spark SQL ne peut lire que depuis Lake Formation tableaux. L'utilisation de Spark SQL pour écrire dans des tables ou créer de nouvelles tables dans Lake Formation n'est pas prise en charge actuellement.
- L'utilisation de Spark SQL pour accéder aux tables Lake Formation qui utilisent le type de données Hive Map n'est pas prise en charge actuellement.
- Lorsque vous utilisez Spark pour accéder aux données stockées dans un format en colonnes, les optimisations de performance telles que le pushdown de prédicats et les lectures vectorisées ne sont pas prises en charge. Les applications Spark SQL qui reposent sur ces optimisations verront leurs performances réduites en cas d'utilisation avec Lake Formation.
- Il n'y a actuellement aucune déconnexion centrale disponible pour les blocs-notes Amazon EMR et Zeppelin.
- Lors de l'utilisation de Spark SQL pour accéder aux données protégées par Lake Formation, AWS CloudTrail les entrées pour l'accès aux données ne contiennent que le nom du rôle IAM associé au Amazon EMR groupe. Elles ne contiennent pas l'utilisateur fédéré qui utilise le bloc-notes.
- La solution de rechange de Spark à HDFS pour la capacité de collecte des statistiques n'est pas prise en charge dans cette version avec Lake Formation. La propriété `spark.sql.statistics.fallBackToHdfs` pour cette fonction est désactivée par défaut. Cette fonction ne fonctionne pas lorsque cette propriété est activée manuellement.
- L'interrogation de tables contenant des partitions sous différents chemins de table dans Amazon S3 n'est actuellement pas prise en charge.
- Il est important de comprendre que l'autorisation au niveau des colonnes Lake Formation empêche les utilisateurs d'accéder aux données des colonnes auxquelles l'utilisateur n'a pas accès. Toutefois, dans certaines situations, les utilisateurs peuvent accéder aux métadonnées décrivant toutes les colonnes de la table, y compris les colonnes auxquelles l'utilisateur n'a pas accès. Les métadonnées de cette colonne sont stockées dans les propriétés de table de la table pour les tables qui utilisent le format de stockage Avro ou des sérialiseurs/désérialiseurs (SerDe) personnalisés dans lesquels le schéma de table est défini dans les propriétés de table avec la définition SerDe. Lorsque vous utilisez Amazon EMR et Lake Formation, nous vous recommandons d'examiner le contenu des propriétés de la table pour les tables que vous protégez et, dans la mesure du possible, de limiter les informations stockées dans les propriétés de la table afin d'empêcher les métadonnées sensibles d'être visibles par les utilisateurs.

Avant de commencer

Avant de lancer un Amazon EMR groupe avec AWS Lake Formation, remplir les conditions préalables suivantes :

- Configurez les rôles IAM AWS et votre fournisseur d'identité pour activer la fédération SAML 2.0.
- Configurer Amazon EMR fonctions de sécurité.

Rubriques

- Présentation des rôles IAM pour Lake Formation (p. 290)

- Configurer la relation d'approbation entre votre fournisseur d'identité et Lake Formation (p. 292)
- Fournisseurs tiers pris en charge pour SAML (p. 294)
- Configuration des fonctions de sécurité EMR (p. 296)

Présentation des rôles IAM pour Lake Formation

L'intégration entre Amazon EMR et AWS Lake Formation repose sur trois rôles clés : le rôle IAM pour Lake Formation, le rôle IAM pour les services AWS et le profil d'instance EC2 pour Amazon EMR. Cette section présente ces rôles et les stratégies que vous devez inclure pour chaque rôle.

Pour plus d'informations sur la configuration des rôles pour Lake Formation, consultez Configurer la relation d'approbation entre votre fournisseur d'identité et Lake Formation (p. 292).

Rôle IAM pour Lake Formation

Le rôle IAM pour Lake Formation définit les privilèges dont disposera un utilisateur se connectant via votre fournisseur d'identité, ainsi que le fournisseur d'identité pouvant assumer ce rôle. Le paramètre `Maximum CLI/API session duration` du rôle définit le délai d'expiration de session pour l'accès aux blocs-notes EMR et à Apache Zeppelin.

- Ce rôle doit être créé avec la stratégie d'autorisations suivante.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "lakeformation:GetDataAccess",
      "lakeformation:GetMetadataAccess",
      "glue:GetUnfiltered*",
      "glue:GetTable",
      "glue:GetTables",
      "glue:GetDatabase",
      "glue:GetDatabases",
      "glue:CreateDatabase",
      "glue:GetUserDefinedFunction",
      "glue:GetUserDefinedFunctions"
    ],
    "Resource": "*"
  }
}
```

> **Note**
>
> N'accordez pas à ce rôle les autorisations pour accéder aux compartiments Amazon S3 gérés par AWS Glue. L'utilisateur fédéré doit accéder aux données via Lake Formation à l'aide de Spark SQL et non directement via Amazon S3.

- Le rôle doit également inclure la stratégie d'approbation suivante, qui permet à votre fournisseur d'identité IAM d'assumer le rôle. Remplacez *`account-id`* avec votre identifiant de compte AWS. Remplacer le *`IAM_identity_provider_name`* avec votre nom de fournisseur d'identité IAM.

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

```
        "Federated": "arn:aws:iam::account-id:saml-provider/IAM_identity_provider_name"
      },
      "Action": "sts:AssumeRoleWithSAML"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com",
          "lakeformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Rôle IAM pour les services AWS

Le rôle IAM pour AWS Services définit les autorisations que Amazon EMR le cluster a accès à l'accès à AWS Lake Formation services. Par exemple, si les tâches en cours d'exécution sur votre cluster nécessitent un accès à Amazon DynamoDB ou à d'autres services AWS, le rôle IAM pour les services AWS doit contenir les stratégies requises pour accéder à ces services. Lorsque vous configurez des stratégies pour ce rôle, assurez-vous que celui-ci n'a pas accès aux opérations d'API suivantes :

- Tout AWS Glue Opérations API.
- Tout AWS Lake Formation Opérations API.
- Tout AWS Security Token Service (STS) `AssumeRole`.
- Accès Amazon S3 aux compartiments gérés par AWS Glue. Le cluster doit accéder aux données via Lake Formation à l'aide de Spark SQL et non directement via Amazon S3.

Profil d'instance EC2

Le profil d'instance EC2 est un type particulier de rôle de service qui définit les autorisations pour que les clusters EMR interagissent avec Lake Formation et d'autres services AWS. Vous pouvez utiliser `EMR_EC2_DefaultRole` lors du lancement d'un cluster ou choisir d'utiliser un profil d'instance EC2 personnalisé. Dans les deux cas, la stratégie suivante doit être ajoutée au rôle comprenant des références au rôle IAM pour Lake Formation et pour les services AWS. Remplacez *`account-id`* avec votre identifiant de compte AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id:role/IAM_Role_For_Lake_Formation"
        },
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::account-id:role/IAM_Role_for_AWS_Services"
        },
        {
            "Effect": "Allow",
            "Action": "lakeformation:GetTemporaryUserCredentialsWithSAML",
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/*"
        }
    ]
}
```

Pour plus d'informations, voir Rôle de service pour les instances EC2 du cluster (profil d'instance EC2) et Personnaliser les rôles IAM.

Configurer la relation d'approbation entre votre fournisseur d'identité et Lake Formation

Pour établir une relation d'approbation entre le fournisseur d'identité (IdP) de votre organisation et AWS, vous devez procéder comme suit :

- Signalez à votre fournisseur d'identité la présence d'AWS en tant que fournisseur de services en ajoutant la relation d'approbation des parties utilisatrices entre le fournisseur d'identité et AWS.
- Indiquez à AWS votre fournisseur d'identité externe en créant un fournisseur d'identité et un rôle IAM pour l'accès SAML dans AWS IAM.

Pour configurer cette relation d'approbation

1. Enregistrez AWS auprès de votre fournisseur d'identité. Le processus d'enregistrement d'AWS auprès de votre fournisseur d'identité dépend du fournisseur que vous utilisez. Pour plus d'informations sur la façon de procéder pour Auth0, Microsoft Active Directory Federation Services (AD FS) et Okta, consultez Fournisseurs tiers pris en charge pour SAML (p. 294).
2. À l'aide de votre fournisseur d'identité, générez un fichier XML de métadonnées pouvant décrire le fournisseur d'identité en tant que fournisseur d'identité IAM dans AWS. Ce fichier doit contenir le nom de l'auteur, une date de création et d'expiration, ainsi que les clés utilisées par AWS pour valider les réponses d'authentification (assertions) de votre organisation. Chaque fournisseur d'identité dispose d'un moyen spécifique d'exporter simplement ces métadonnées. Pour plus d'informations, consultez la documentation de votre fournisseur d'identité.

   Vous devez télécharger le fichier XML de métadonnées sur un Amazon S3 seau. Lorsque vous lancez un cluster qui s'intègre à Lake Formation, vous devez spécifier le chemin d'accès au compartiment S3.
3. Dans la console IAM, créez une entité de fournisseur d'identité SAML.
   a. Connectez-vous à AWS Management Console et ouvrez la console IAM; à l'adresse https://console.aws.amazon.com/iam/.
   b. Dans le volet de navigation, choisissez Fournisseurs d'identité, Créer un fournisseur.
   c. Pour Type de fournisseur, choisissez Choisir un type de fournisseur, Saml.
   d. Entrez un nom pour le fournisseur d'identité.
   e. Pour Document de métadonnées, cliquez Choisir un fichier, spécifiez le document de métadonnées SAML que vous avez téléchargé depuis votre idp à l'étape précédente, puis choisissez Ouvert.
   f. Vérifiez les informations que vous avez fournies, puis cliquez sur Créer.
4. Dans la console IAM, créez un rôle IAM pour la fédération d'identité.
   a. Dans le volet de navigation de la console IAM, choisissez Rôles, Créer un rôle.
   b. Choisissez le Fédération SAML 2.0 type de rôle.
   c. Pour Fournisseur SAML, choisissez le fournisseur pour votre rôle.
   d. Choisir Autoriser l'accès au programmatique et à la console de gestion AWS pour créer un rôle qui peut être supposé par programmation et de la console.
   e. Examinez vos informations de confiance SAML 2.0, puis choisissez Suivant : Permissions.

f. Créez la stratégie d'autorisations pour le rôle en fonction de l'exemple dans Présentation des rôles IAM pour Lake Formation (p. 290).
g. Choisir Suivant : Tags (Balises) .
h. Choisir Suivant : Vérifiez .
i. Pour Nom du rôle, tapez un nom de rôle. Les noms de rôle doivent être uniques dans votre compte AWS.
j. Passez en revue le rôle puis choisissez Créer un rôle.
k. Cliquez sur l'onglet Roles (Rôles), recherchez le nom du rôle créé lors de la dernière étape.
l. Choisir Relations de confiance, puis sélectionnez Modifier la relation de confiance.
m. Remplacer le document de politique existant par le rôle IAM pour Lake Formation politique de fiducie spécifiée dans le Présentation des rôles IAM pour Lake Formation (p. 290). Puis cliquez sur Mettre à jour la relation de confiance.

5. Dans le fournisseur d'identité de votre organisation, vous devez configurer les assertions SAML qui mappent les utilisateurs de votre organisation vers le fournisseur d'identité et le rôle IAM pour Lake Formation qui vient d'être créé. Pour ce faire, configurez les trois éléments d'attribut présentés dans le tableau suivant.
   - Remplacez *account-id* avec votre identifiant de compte AWS.
   - Remplacez *IAM_Role_For_Lake_Formation* avec le nom du rôle IAM pour Lake Formation que vous avez créé.
   - Remplacez *IAM_identity_provider_name* avec le nom du fournisseur d'identité IAM que vous avez créé dans les étapes précédentes.
   - Remplacez *user_alias* avec le nom de l'attribut utilisé pour maintenir le nom d'utilisateur défini dans votre organisation.

Éléments d'attribut	Value
https://aws.amazon.com/SAML/Attributes/Role	arn :aws :iam :*account-id*rôle*IAM_Role_For_Lake_Formation*,arn :aws :iam :*account-id*saml-fournisseur/*IAM_identity_provider_name*
https://aws.amazon.com/SAML/Attributes/RoleSessionName	*user_alias*
https://lakeformation.amazon.com/SAML/Attributes/Username	*user_alias*

La procédure exacte pour leur mappage dépend du fournisseur d'identité que vous utilisez. Pour plus d'informations, voir la section suivante, Fournisseurs tiers pris en charge pour SAML (p. 294).

Pour plus d'informations, voir Configuration des assertions SAML pour la réponse d'authentification.

> **Note**
>
> Un utilisateur peut obtenir des privilèges de formation du lac pour tous les groupes appartenant à. La liste des groupes auxquels un utilisateur appartient peut être envoyée dans un attribut SAML personnalisé spécifique à la formation du lac, comme l'exemple suivant montre.

```
<AttributeStatement>
<Attribute Name="https://lakeformation.amazon.com/SAML/Attributes/Groups">
<AttributeValue>group1</AttributeValue>
<AttributeValue>group2</AttributeValue>
</Attribute>
```

```
</AttributeStatement>
```

Fournisseurs tiers pris en charge pour SAML

Intégration entre Amazon EMR et AWS Lake Formation prend en charge la fédération SAML 2.0 avec les fournisseurs tiers suivants : Microsoft Active Directory Federation Services (AD FS), Auth0 et Okta. Les sections suivantes fournissent des informations pour vous aider à configurer ces sauts pour travailler avec AWS Lake Formation fédération.

Auth0

Intégration AWS dans Auth0 – Cette page du site Web de documentation Auth0 explique comment configurer une authentification unique (SSO) avec AWS Management Console. Elle inclut également un exemple JavaScript.

Pour activer l'accès fédéré à Lake Formation, personnalisez les étapes suivantes dans la documentation Auth0 :

- Lorsque vous fournissez une URL de rappel d'application, indiquez une URL temporaire, comme illustré dans l'exemple suivant. Mise à jour: *public-dns* avec le nom DNS réel pour votre noeud principal après avoir lancé votre cluster.

```
https://public-dns:8442/gateway/knoxsso/api/v1/websso?
pac4jCallback=true&client_name=SAML2Client
```

- Lors de la configuration de SAML, coller le code de configuration SAML suivant dans Paramètres.

```
{
    "audience": "urn:amazon:webservices",
    "mappings": {
        "email": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
        "name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
    },
    "createUpnClaim": false,
    "passthroughClaimsWithNoMapping": false,
    "mapUnknownClaimsAsIs": false,
    "mapIdentities": false,
    "nameIdentifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent",
    "nameIdentifierProbes": [
        "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
    ]
}
```

- Lorsque vous mappez le rôle AWS à un utilisateur, créez une règle avec le code suivant. Remplacer le *IAM_Role_For_Lake_Formation* avec le nom du rôle IAM pour Lake Formation que vous avez créé. Remplacer le *IAM_identity_provider_name* avec le nom du fournisseur d'identité IAM que vous avez créé pour Auth0.

```
function (user, context, callback) {
  user.awsRole = 'arn:aws:iam::account-
id:role/IAM_Role_For_Lake_Formation,arn:aws:iam::account-id:saml-
provider/IAM_identity_provider_name';
  // the username must not contain "@" - as it is not a valid Linux username
  user.glueUser = user.name.replace(/@.*/, '');

  context.samlConfiguration.mappings = {
    'https://aws.amazon.com/SAML/Attributes/Role': 'awsRole',
    'https://aws.amazon.com/SAML/Attributes/RoleSessionName': 'glueUser',
    'https://lakeformation.amazon.com/SAML/Attributes/Username': 'glueUser'
  };
```

```
  callback(null, user, context);
}
```

Microsoft Active Directory Federation Services (AD FS)

Authentification fédérée AWS avec Active Directory Federation Services (AD FS) – Cette publication sur le blog de sécurité AWS montre comment configurer AD FS et activer la fédération SAML avec AWS.

Pour activer l'accès fédéré à Lake Formation, personnalisez les étapes suivantes :

- Pour ajouter une relation d'approbation des parties utilisatrices, entrez manuellement les données relatives aux parties utilisatrices au lieu d'importer des métadonnées à partir de l'URL existante. Sélectionnez le Permettre à tous les utilisateurs d'accéder à cette partie de confiance. Pour l'URL approuvée du point de terminaison, indiquez une URL temporaire, comme illustré dans l'exemple suivant. Mise à jour: *public-dns* avec le nom DNS réel pour votre noeud principal après avoir lancé votre cluster.

```
https://public-dns:8442/gateway/knoxsso/api/v1/websso?
pac4jCallback=true&client_name=SAML2Client
```

- À l'étape de Modifier la politique d'émission de réclamation, personnaliser les trois règles `NameId`, `RoleSessionName`, et `Role` basé sur les valeurs des éléments d'attribut dans Configurer la relation d'approbation entre votre fournisseur d'identité et Lake Formation (p. 292).

Okta

Configurer une application SAML dans Okta – Sur cette page du site d'assistance Okta, vous pouvez apprendre à configurer Okta en fournissant des métadonnées au sujet de la partie de confiance.

Pour activer l'accès fédéré à Lake Formation, personnalisez les étapes suivantes :

- Lors de la configuration de SAML, pour URL de connexion unique, utilisez l'URL temporaire, comme illustré dans l'exemple suivant. Mettre à jour le *public-dns* avec le nom DNS réel pour votre noeud principal après avoir lancé votre cluster.

```
https://public-dns:8442/gateway/knoxsso/api/v1/websso?
pac4jCallback=true&client_name=SAML2Client
```

- Pour le URI public (ID entité SP) , remplir `urn:amazon:webservices`.
- Dans le Énoncés d'attributs ajouter trois énoncés d'attributs comme démontré dans la procédure suivante. Remplacer le *IAM_Role_For_Lake_Formation* avec le nom du rôle IAM pour Lake Formation que vous avez créé. Remplacer le *IAM_identity_provider_name* avec le nom du fournisseur d'identité IAM que vous avez créé dans les étapes précédentes. Remplacez *user_alias* avec le nom de l'attribut utilisé pour maintenir le nom d'utilisateur défini dans votre organisation.
  1. Nom = `https://aws.amazon.com/SAML/Attributes/Role`

     Valeur : `arn:aws:iam::`*account-id*`:role/`*IAM_Role_For_Lake_Formation*`,arn:aws:iam::`*account-id*`:saml-provider/`*IAM_identity_provider_name*
  2. Nom = `https://aws.amazon.com/SAML/Attributes/RoleSessionName`

     Valeur : *user_alias*
  3. Nom = `https://lakeformation.amazon.com/SAML/Attributes/Username`

     Valeur : *user_alias*

Configuration des fonctions de sécurité EMR

Pour garantir que Amazon EMR est intégré en toute sécurité avec AWS Lake Formation, configurez les fonctions de sécurité EMR suivantes :

- Activez l'authentification Kerberos à l'aide du KDC dédié au cluster. Pour plus d'informations, voir Utiliser l'authentification Kerberos.
- Configurer votre Amazon EC2 groupe de sécurité ou Amazon VPC la liste de contrôle d'accès réseau (ACL) pour permettre l'accès à l'agent proxy (port 8442) depuis les postes de travail de votre utilisateur. Pour plus d'informations, voir Contrôle du trafic réseau avec les groupes de sécurité.
- (Facultatif) Activez le chiffrement en transit ou au repos. Pour plus d'informations, voir Options de chiffrement dans le Amazon EMR Guide de gestion.
- (Facultatif) Créez une paire de clés TLS (Transport Layer Security) personnalisée pour l'agent proxy. Pour plus d'informations, consultez Personnalisation du certificat de l'agent proxy (p. 300).

Pour plus d'informations, voir Sécurité dans Amazon EMR.

Lancez un cluster Amazon EMR avec Lake Formation

Cette section fournit des informations sur la manière de lancer un cluster Amazon EMR intégré à Lake Formation. Il montre également comment mettre à jour l'URL d'authentification unique dans votre fournisseur d'identité, utiliser les blocs-notes avec Lake Formation et personnaliser le certificat d'agent proxy.

Pour plus d'informations sur le dépannage des problèmes courants, voir Dépannage d'un cluster de formation du lac dans le Amazon EMR Guide de gestion.

Rubriques

- Lancer un Amazon EMR groupe avec Lake Formation à l'aide de la console (p. 296)
- Lancer un Amazon EMR cluster intégré avec Lake Formation utilisation de la CLI (p. 297)
- Mise à jour de l'URL de rappel ou d'authentification unique dans le fournisseur d'identité (p. 298)
- Utilisation des blocs-notes Lake Formation (p. 299)
- Personnalisation du certificat de l'agent proxy (p. 300)

Lancer un Amazon EMR groupe avec Lake Formation à l'aide de la console

1. Créer une configuration de sécurité qui spécifie le AWS Lake Formation intégration option :
   1. Dans le Amazon EMR console, sélectionner Configurations de sécurité, Créer.
   2. Type A Nom pour la configuration de sécurité. Ce nom est utilisé pour spécifier la configuration de sécurité lorsque vous créez un cluster.
   3. Sous AWS Lake Formation intégration, sélectionnez Activer le contrôle d'accès à grain fin géré par AWS Lake Formation.
   4. Sélectionner un Rôle IAM pour AWS Lake Formation pour appliquer.

      Note

      Pour plus d'informations, consultez Présentation des rôles IAM pour Lake Formation (p. 290).
   5. Sélectionner un Rôle IAM pour les autres services AWS pour appliquer.

6. Chargez les métadonnées de votre fournisseur d'identité (IdP) en spécifiant le chemin d'accès S3 où se trouvent les métadonnées.

   Note

   Pour plus d'informations, consultez Configurer la relation d'approbation entre votre fournisseur d'identité et Lake Formation (p. 292).

7. Configurer d'autres options de configuration de sécurité selon les besoins et choisir Créer. Vous devez activer l'authentification Kerberos à l'aide du KDC dédié au cluster. Pour plus d'informations, consultez Configuration des fonctions de sécurité EMR (p. 296).

2. Lancez un cluster avec la configuration de sécurité que vous avez spécifiée à l'étape précédente. Pour plus d'informations, voir Spécifier une configuration de sécurité pour un cluster.

Lancer un Amazon EMR cluster intégré avec Lake Formation utilisation de la CLI

La procédure suivante montre comment lancer un cluster Amazon EMR avec Zeppelin intégré à AWS Lake Formation.

1. Créez un fichier `security-configuration.json` pour la configuration de sécurité avec le contenu suivant.

   - Spécifiez le chemin d'accès complet au fichier de métadonnées du fournisseur d'identité dans S3.
   - Remplacez *`account-id`* avec votre identifiant de compte AWS.
   - Spécifiez une valeur pour `TicketLifetimeInHours` afin de déterminer la période pendant laquelle un ticket Kerberos émis par le KDC est valide.

```
{
    "LakeFormationConfiguration": {
        "IdpMetadataS3Path": "s3:///mybucket/myfolder/idpmetadata.xml",
        "EmrRoleForUsersARN": "arn:aws:iam::account-id:role/IAM_Role_For_AWS_Services",
        "LakeFormationRoleForSAMLPrincipalARN": "arn:aws:iam::account-
id:role/IAM_Role_For_Lake_Formation"
    },
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24
            }
        }
    }
}
```

2. Exécutez la commande suivante pour créer une configuration de sécurité.

```
aws emr create-security-configuration \
--security-configuration file:///./security-configuration.json \
--name security-configuration
```

3. Créez un fichier `configurations.json` qui configure le metastore Hive.

```
[
    {
        "Classification": "spark-hive-site",
        "Properties": {
            "hive.metastore.glue.catalogid": "account-id"
```

```
        }
    }
]
```

4. Exécutez la commande suivante pour lancer un Amazon EMR groupe.
   - Remplacez *subnet-00xxxxxxxxxxxxx11* avec votre ID de sous-réseau.
   - Remplacer le *EC2_KEY_PAIR* avec le nom de votre paire de clés EC2 pour ce groupe. La paire de clés EC2 est facultative et obligatoire uniquement si vous souhaitez utiliser SSH pour accéder à votre cluster.
   - Remplacez *cluster-name* avec le nom de votre cluster.

```
aws emr create-cluster --region us-east-1 \
--release-label emr-5.26.0 \
--use-default-roles \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.xlarge \
InstanceGroupType=CORE,InstanceCount=1,InstanceType=m4.xlarge \
--applications Name=Zeppelin Name=Livy \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyClusterKDCAdminPassword \
--configurations file://confugurations.json \
--ec2-attributes KeyName=EC2_KEY_PAIR,SubnetId=subnet-00xxxxxxxxxxxx11 \
--security-configuration security-configuration \
--name cluster-name
```

Mise à jour de l'URL de rappel ou d'authentification unique dans le fournisseur d’identité

1. Recherchez l'adresse IP publique du nœud principal et l'ID d'instance principal dans votre cluster à l'aide de la console ou de l'interface de ligne de commande.
2. Configurez une URL de rappel dans votre compte de fournisseur d’identité :
   - Lorsque vous utilisez AD FS comme fournisseur d’identité, effectuez les opérations suivantes :
     1. Depuis la console de gestion AD FS, allez à Confiance envers les parties.
     2. Cliquez avec le bouton droit de la souris sur le nom d’affichage de la confiance de votre partie de réponse, puis choisissez Propriétés.
     3. Dans le Propriétés , choisissez le Terminaux.
     4. Sélectionnez l’URL temporaire que vous avez fournie précédemment, puis choisissez modifier.
     5. Dans le Modifier endpoint , mettez à jour l’URL approuvée avec le nom DNS correct pour votre noeud principal.
     6. Dans le Ajouter un point de terminaison fenêtre, remplissez le URL approuvée avec votre DNS public de noeud maître. Par exemple,

        ```
        https://ec2-11-111-11-111.compute-1.amazonaws.com:8442/gateway/knoxsso/api/v1/
        websso?pac4jCallback=true&client_name=SAML2Client
        ```

     7. Choisir OK.
   - Lorsque vous utilisez Auth0 comme fournisseur d'identité, effectuez les opérations suivantes :
     1. Accédez à https://auth0.com/ et connectez-vous.
     2. Dans le panneau de gauche, choisissez Applications.
     3. Sélectionnez votre application précédemment créée.

4. Sur le Paramètres onglet, mise à jour URL de rappel autorisées avec votre DNS public de noeud principal.

- Lorsque vous utilisez Okta comme fournisseur d'identité, effectuez les opérations suivantes :
  1. Accédez à https://developer.okta.com/ et connectez-vous.
  2. Dans le coin supérieur droit, choisissez Admin, puis choisissez le Applications.
  3. Sélectionnez le nom de votre application.
  4. Sur le Généralités sous le nom de votre application, choisissez Paramètres SAML, puis choisissez Modifier.
  5. Sur le Configurer SAML onglet, mise à jour Authentification unique sur URL avec votre DNS public de noeud principal.

Utilisation des blocs-notes Lake Formation

Les blocs-notes Apache Zeppelin et EMR sont intégrés à Lake Formation et peuvent être utilisés une fois qu'un cluster EMR intégré à Lake Formation a été créé.

Pour accéder aux deux applications de bloc-notes, vous devez tout d'abord vous assurer que le groupe de sécurité EC2 ou la liste de contrôle d'accès (ACL) réseau VPC de votre cluster est configuré pour autoriser l'accès à l'agent proxy (port 8442) depuis votre bureau.

> Note
>
> L'agent proxy sur le cluster EMR utilise un certificat TLS (Transport Layer Security) auto-signé par défaut ; votre navigateur vous invite à accepter le certificat avant de continuer. Si vous souhaitez utiliser un certificat personnalisé pour l'agent proxy, consultez la section relative à la personnalisation du certificat de l'agent proxy.

Apache Zeppelin

Pour accéder à Apache Zeppelin, utiliser la console EMR pour localiser le *`Master public DNS`* du cluster Résumé. À l'aide de votre navigateur, accédez à https ://*`MasterPublicDNS`*: 8442/gateway/default/zeppelin/. Assurez-vous que l'URL inclut la barre oblique de fin.

Une fois le certificat de l'agent proxy accepté, votre navigateur vous redirige vers votre fournisseur d'identité (IdP) pour vous authentifier. Une fois authentifié, vous serez redirigé vers Zeppelin.

Création de votre premier bloc-notes Zeppelin

Pour commencer, créez un nouveau carnet en sélectionnant Carnet, Créer une nouvelle note. Donnez un nom à votre notebook et utilisez la valeur par défaut livy interprète.

Pour afficher la liste des bases de données Lake Formation, utilisez la commande SQL Spark suivante.

```
spark.sql("show databases").show()
```

Pour interroger une table Lake Formation spécifique, utilisez la commande SQL Spark suivante. Remplacez *`database.table`* avec des bases de données et des tableaux réels dans la formation du lac :

```
spark.sql("SELECT * FROM database.table limit 10").show()
```

Blocs-notes EMR

Les blocs-notes EMR peuvent être créés à l'aide de la console Amazon EMR et utilisés avec un cluster EMR existant intégré à Lake Formation.

Pour créer un bloc-notes EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisir Carnets, Créer un carnet.
3. Saisissez un Nom du carnet et une option Description du carnet.
4. Sélectionner Choisir un cluster existant, puis Choisir.
5. Sélectionnez un cluster EMR existant intégré à Lake Formation.
6. Sélectionner Créer un carnet pour créer le carnet.

Une fois le carnet de notes créé, sélectionnez le carnet et cliquez sur Ouvert. Vous serez redirigé vers l'agent proxy sur le Amazon EMR groupe. Une fois que vous avez accepté le certificat de l'agent proxy, votre navigateur vous redirige vers votre fournisseur d'identité (IdP) pour vous authentifier. Une fois authentifié, vous serez redirigé vers le bloc-notes EMR.

Pour plus d'informations, voir Utilisation des carnets d'adresses électroniques Amazon dans le Amazon EMR Guide de gestion.

Personnalisation du certificat de l'agent proxy

L'agent proxy utilise un certificat TLS (Transport Layer Security) auto-signé par défaut. Pour utiliser un certificat personnalisé pour l'agent proxy, vous devez tout d'abord obtenir un certificat, une chaîne de certificats et une clé privée auprès de votre autorité de certification. Avec ces éléments, un fichier PKCS12 sera utilisé pour protéger les éléments de clé afin qu'ils puissent être importés dans le magasin de clés de l'agent proxy. L'agent proxy est basé sur Apache Knox. Vous pouvez remplacer le certificat par défaut par votre certificat personnalisé en procédant comme suit.

Dans les étapes suivantes, remplacer *`MasterPublicDNS`* avec la valeur qui apparaît pour DNS public principal sur le Résumé onglet du détails du cluster. Par exemple, *`ec2-11-222-33-44.compute-1.amazonaws.com`*.

1. Pour créer un fichier PKCS12 à partir de votre certificat, de votre chaîne de certificats et de votre clé privée, exécutez la commande suivante sur un hôte avec les fichiers de certificat et `openssl` installés.

   ```
   openssl pkcs12 -export -out proxy_agent_certificate.pfx -inkey private.key -
   in certificate.cer -certfile certchain.cer
   ```

2. Copier le `proxy_agent_certificate.pfx` fichier à `/home/hadoop` répertoire du noeud principal de votre cluster.

   ```
   scp -i EC2KeyPair.pem proxy_agent_certificate.pfx hadoop@MasterPublicDNS:/home/hadoop
   ```

3. SSH dans le nœud principal de votre cluster.

   ```
   ssh -i EC2KeyPair.pem hadoop@MasterPublicDNS
   ```

4. Recherchez votre clé principale spécifique au cluster à l'aide de la commande suivante.

   ```
   less /etc/knox/conf/gateway-site.xml
   ```

   Recherchez le *`gateway.master.secret`* propriété et copier le contenu du `value` comme vous le souhaitez pour les prochaines étapes.

5. Créez une copie de sauvegarde des magasins de clés d'agent proxy existants à l'aide des commandes suivantes.

   ```
   sudo -s
   cd /mnt/var/lib/knox/data/security/keystores
   ```

```
mkdir backups
mv gateway.jks __gateway-credentials.jceks backups/
```

6. Importez vos certificats personnalisés dans un nouveau magasin de clés à l'aide des commandes suivantes.

```
sudo -s
cd /mnt/var/lib/knox/data/security/keystores
keytool -importkeystore \
-srckeystore /home/hadoop/proxy_agent_certificate.pfx \
-srcstoretype pkcs12 -destkeystore gateway.jks \
-deststoretype jks \
-srcalias 1 \
-destalias gateway-identity
```

   Lorsque vous êtes invité à `Enter destination keystore password`, utilisez le secret principal Knox de la `gateway-site.xml` fichier.

   Assurez-vous que le fichier `gateway.jks` nouvellement créé appartient à l'utilisateur knox à l'aide de la commande suivante.

```
chown knox:knox gateway.jks
```

   Si votre clé privée est protégée par un mot de passe, assurez-vous que Knox connait ce mot de passe.

```
sudo -u knox bash
cd /usr/lib/knox
bin/knoxcli.sh create-cert create-alias gateway-identity-passphrase
```

   Entrez le mot de passe protégeant votre clé privée lorsque vous y êtes invité.

7. Redémarrez Knox à l'aide des commandes suivantes :

```
sudo -u knox bash
cd /usr/lib/knox
bin/gateway.sh stop
```

   Knox doit être redémarré automatiquement. Vous pouvez vérifier le statut de Knox en consultant `/var/log/knox/gateway.log`.

8. Pour vous assurer que le nouveau certificat est utilisé par l'agent proxy, allez à Apache Zeppelin https ://*`MasterPublicDNS`*: 8442/gateway/default/zeppelin/. Vous pouvez utiliser votre navigateur pour inspecter le certificat afin de vous assurer qu'il s'agit de votre certificat personnalisé.

Contrôle du trafic réseau avec des groupes de sécurité

Les groupes de sécurité agissent en tant que pares-feux virtuels pour vos instances EC2 dans votre cluster afin de contrôler le trafic entrant et sortant. Chaque groupe de sécurité dispose d'un ensemble de règles qui contrôle le trafic entrant et un ensemble distinct de règles pour contrôler le trafic sortant. Pour plus d'informations, consultez Groupes de sécurité Amazon EC2 pour les instances Linux dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Vous utilisez deux classes de groupes de sécurité Amazon EMR : les groupes de sécurité gérés par Amazon EMR et des groupes de sécurité supplémentaires.

Chaque cluster dispose de groupes de sécurité qui lui sont associés. Vous pouvez utiliser les groupes de sécurité gérés par défaut ou spécifier des groupes de sécurité gérés personnalisés. Dans les deux cas, Amazon EMR ajoute automatiquement des règles aux groupes de sécurité gérés dont un cluster a besoin pour communiquer entre les instances de cluster et les services AWS.

Les groupes de sécurité supplémentaires sont facultatifs. Vous pouvez les spécifier en plus des groupes de sécurité gérés pour adapter l'accès aux instances de cluster. Les groupes de sécurité supplémentaires contiennent uniquement des règles que vous définissez. Amazon EMR ne les modifie pas.

Les règles qu'Amazon EMR crée dans les groupes de sécurité gérés autorisent le cluster à communiquer uniquement entre les composants internes. Pour autoriser l'accès des utilisateurs et des applications à un cluster depuis l'extérieur du cluster, vous pouvez modifier les règles gérées dans les groupes de sécurité, créer des groupes de sécurité supplémentaires avec des règles supplémentaires, ou les deux.

> **Important**
>
> La modification des règles dans les groupes de sécurité gérés peut engendrer des conséquences imprévues. Vous pouvez malencontreusement bloquer le trafic requis pour le bon fonctionnement des clusters et provoquer des erreurs, car les nœuds sont inaccessibles. Planifiez et testez soigneusement les configurations de groupe de sécurité avant la mise en œuvre.
>
> **Warning**
>
> La valeur par défaut du groupe de sécurité géré par EMR pour l'instance principale dans les sous-réseaux publics, `ElasticMapReduce-master`, est préconfigurée avec une règle qui autorise le trafic entrant sur le port 22 à partir de toutes les sources (IPv4 0.0.0.0/0). Cela permet de simplifier les connexions clients SSH initiales au nœud maître. Nous vous recommandons vivement de modifier cette règle entrante afin de restreindre le trafic uniquement aux sources fiables ou de spécifier un groupe de sécurité personnalisé qui limite l'accès. Pour plus d'informations, consultez la section Utilisation des groupes de sécurité gérés par Amazon EMR (p. 303).

Vous pouvez spécifier les groupes de sécurité uniquement lorsque vous créez un cluster. Ils ne peuvent pas être ajoutés à un cluster ou à des instances de cluster pendant qu'un cluster est en cours d'exécution, mais vous pouvez modifier, ajouter et supprimer des règles des groupes de sécurité existants. Les règles prennent effet dès que vous les enregistrez.

Par défaut, les groupes de sécurité sont restrictifs. Le trafic est rejeté, sauf si une règle autorisant le trafic est ajoutée. S'il y a plus d'une règle qui s'applique au même trafic et à la même source, c'est la règle la plus permissive qui s'applique. Par exemple, si vous avez une règle qui autorise SSH 192.0.2.12/32 à partir de l'adresse IP et une autre règle qui autorise l'accès à tout le trafic TCP à partir de la plage 192.0.2.0/24, la règle qui autorise tout le trafic TCP à partir de la plage qui inclut 192.0.2.12 est prioritaire. Dans ce cas, le client à 192.0.2.12 peut avoir plus d'accès que vous n'auriez souhaité.

Soyez vigilant lorsque vous modifiez les règles du groupe de sécurité. Assurez-vous d'ajouter des règles qui autorisent uniquement le trafic provenant de clients approuvés pour les protocoles et les ports qui sont requis. Nous déconseillons les règles entrantes qui autorisent l'accès public, c'est-à-dire le trafic provenant de sources spécifiées comme IPv4 0.0.0.0/0 ou IPv6::/0. Vous pouvez configurer le blocage d'accès public Amazon EMR dans chaque région que vous utilisez, pour empêcher la création d'un cluster, si une règle autorise l'accès public sur n'importe quel port que vous n'ajoutez pas à une liste d'exceptions. Le port 22 pour SSH figure dans la liste des exceptions par défaut. Pour les comptes AWS créés après juillet 2019, le blocage d'accès public Amazon EMR est activé par défaut. Pour AWS les comptes qui ont créé un cluster avant juillet 2019, le blocage d'accès public Amazon EMR est désactivé par défaut. Pour plus d'informations, consultez la section Utilisation d'Amazon EMR Block Public Access (p. 312).

Rubriques

- Utilisation des groupes de sécurité gérés par Amazon EMR (p. 303)
- Utilisation des groupes de sécurité supplémentaires (p. 309)

- Spécification de groupes gérés et supplémentaires Amazon EMR (p. 309)
- Spécification des groupes de sécurité EC2 pour les blocs-notes EMR (p. 311)
- Utilisation d'Amazon EMR Block Public Access (p. 312)

Utilisation des groupes de sécurité gérés par Amazon EMR

Les différents groupes de sécurité gérés sont associés à l'instance principale et aux instances principales et de tâche dans un cluster. Un groupe de sécurité géré supplémentaire pour l'accès au service est obligatoire lorsque vous créez un cluster dans un sous-réseau privé. Pour plus d'informations sur le rôle des groupes de sécurité gérés par rapport à votre configuration réseau, consultez Options Amazon VPC (p. 138).

Lorsque vous spécifiez des groupes de sécurité gérés pour un cluster, vous devez utiliser le même type de groupe de sécurité par défaut ou personnalisé pour tous les groupes de sécurité gérés. Par exemple, vous ne pouvez pas spécifier un groupe de sécurité personnalisé pour l'instance principale, puis ne pas spécifier un groupe de sécurité personnalisé pour les instances principales et de tâches.

Si vous utilisez les groupes de sécurité gérés par défaut, vous n'avez pas besoin de les spécifier lorsque vous créez un cluster. Amazon EMR utilise automatiquement les valeurs par défaut. En outre, si les valeurs par défaut n'existent pas encore dans le VPC du cluster, Amazon EMR les crée. Amazon EMR les crée également si vous les spécifiez explicitement et qu'ils n'existe pas encore.

Vous pouvez modifier les règles dans les groupes de sécurité gérés après que les clusters soient créés. Lorsque vous créez un cluster, Amazon EMR vérifie les règles dans les groupes de sécurité gérés que vous spécifiez, puis crée les règles entrant aux règles dont le nouveau cluster a besoin en plus des règles qui peuvent avoir été ajoutées précédemment. Sauf indication contraire, chaque règle pour les groupes de sécurité gérés par EMR par défaut est également ajoutée aux groupes de sécurité gérés par EMR personnalisés que vous spécifiez.

Les groupes de sécurité gérés par défaut sont les suivants :

- ElasticMapReduce-master

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux publics) (p. 304).
- ElasticMapReduce-slave

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale et de tâches (sous-réseaux publics) (p. 306).
- ElasticMapReduce-Master-Private

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux privés) (p. 306).
- ElasticMapReduce-Slave-Private

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour les instances principales et de tâches (sous-réseaux privés) (p. 307).
- ElasticMapReduce-ServiceAccess

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'accès au service (sous-réseaux privés) (p. 308).

Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux publics)

Le groupe de sécurité géré par défaut pour l'instance principale dans les sous-réseaux publics a le Nom du groupe sur Maître ElasticMapReduce. Il a les règles suivantes : Si vous spécifiez un groupe de sécurité géré personnalisé, Amazon EMR ajoutera toutes les mêmes règles à l'exception de la règle SSH.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Règles entrantes				
Tous les ICMP-IPv4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré pour l'instance principale. En d'autres termes, le même groupe de sécurité dans lequel la règle s'affiche.	Ces règles réflexives autorisent le trafic entrant à partir de toute instance associée au groupe de sécurité spécifié. L'utilisation de la valeur par défaut `ElasticMapReduce-master` pour plusieurs clusters autorise les nœuds principaux et de tâches de ces clusters à communiquer entre eux sur ICMP ou sur n'importe quel port TCP ou UDP. Spécifiez les groupes de sécurité gérés personnalisés pour restreindre l'accès inter-cluster.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré spécifié pour les nœuds principaux et de tâches.	Ces règles autorisent tout le trafic ICMP entrant et le trafic sur n'importe quel port TCP ou UDP à partir de toutes les instances principales et de tâches qui sont associées au groupe de sécurité spécifié, même si les instances se trouvent dans des clusters différents.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
Personnalisée	TCP	8443	Différentes plages d'adresses IP Amazon	Ces règles permettent au gestionnaire de cluster de communiquer avec le nœud principal.
SSH	TCP	22	0.0.0.0/0	Autorisez les connexions SSH entrantes en provenance de n'importe quelle source. Warning Nous vous recommandons vivement de modifier cette règle entrante afin de restreindre le trafic uniquement à partir de sources fiables. Vous pouvez également spécifier un groupe de sécurité géré par EMR personnalisé pour limiter l'accès. Pour de plus amples informations, veuillez consulter procedure to remove public access to SSH (p. 304).

Pour supprimer la règle entrante qui autorise l'accès public via SSH pour le groupe de sécurité ElasticMapReduce-master

La procédure suivante suppose que le groupe de sécurité ElasticMapReduce-master n'a pas encore été modifié. De plus, pour modifier les groupes de sécurité, vous devez être connecté à AWS en tant qu'utilisateur racine ou principal IAM autorisé à gérer les groupes de sécurité pour le VPC dans lequel figure le cluster. Pour plus d'informations, consultez Modification des autorisations pour un utilisateur IAM et l'exemple de stratégie qui permet de gérer les groupes de sécurité EC2 dans le IAM Guide de l'utilisateur.

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Clusters.
3. Choisissez le Name (Nom) du cluster.
4. Sous Security and access (Sécurité et accès), choisissez le lien Security groups for Master (Groupes de sécurité pour le principal).

5. Choisissez ElasticMapReduce-master dans la liste.
6. Choisissez Entrant, Modifier.
7. Recherchez la règle dotée des paramètres suivants et choisissez l'icône x pour la supprimer :
   - Type

     SSH
   - Port

     22
   - Source

     Personnalisé 0.0.0.0/0
8. Faites défiler l'écran jusqu'au bas de la liste de règles et choisissez Add rule (Ajouter une règle).
9. Dans Type, sélectionnez SSH.

Le champ Protocol (Protocole) est automatiquement défini sur TCP et Port Range (Plage de ports) sur 22.

10. Pour la source, sélectionnez My IP (Mon IP).

    Cette action ajoute automatiquement l'adresse IP de votre ordinateur client comme adresse source. Sinon, vous pouvez ajouter une plage d'adresses IP de clients approuvés Custom (Personnalisée) et choisir Add rule (Ajouter une règle) pour créer des règles supplémentaires pour d'autres clients. Dans de nombreux environnements de réseau, les adresses IP sont attribuées dynamiquement. Par conséquent, vous devrez peut-être modifier régulièrement les règles de groupe de sécurité pour mettre à jour l'adresse IP de confiance des clients.
11. Choisissez Save.
12. Choisissez éventuellement ElasticMapReduce-slave dans la liste et répétez les étapes ci-dessus pour autoriser le client SSH à accéder aux nœuds principaux et de tâches des clients de confiance.

Groupe de sécurité géré par Amazon EMR pour l'instance principale et de tâches (sous-réseaux publics)

Le groupe de sécurité géré par défaut pour les instances principales et de tâches dans les sous-réseaux publics possède le Group Name (Nom de groupe) ElasticMapReduce-master. Le groupe de sécurité géré par défaut comporte les règles suivantes et Amazon EMR ajoute les mêmes règles si vous spécifiez un groupe de sécurité géré personnalisé.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Règles entrantes				
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré spécifié pour les instances principales et de tâches. En d'autres termes, le même groupe de sécurité dans lequel la règle s'affiche.	Ces règles réflexives autorisent le trafic entrant à partir de toute instance associée au groupe de sécurité spécifié. Utilisation de la valeur par défaut `ElasticMapReduce-slave` pour plusieurs clusters autorise les instances principales et de tâches de ces clusters à communiquer entre elles sur ICMP ou sur n'importe quel port TCP ou UDP. Spécifiez les groupes de sécurité gérés personnalisés pour restreindre l'accès inter-cluster.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré pour l'instance principale.	Ces règles autorisent tout le trafic ICMP entrant et le trafic sur n'importe quel port TCP ou UDP à partir de toutes les instances principales qui sont associées au groupe de sécurité spécifié, même si les instances se trouvent dans des clusters différents.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		

Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux privés)

Le groupe de sécurité géré par défaut pour l'instance principale dans les sous-réseaux privés possède le Group Name (Nom de groupe) ElasticMapReduce-master. Le groupe de sécurité géré par défaut comporte

les règles suivantes et Amazon EMR ajoute les mêmes règles si vous spécifiez un groupe de sécurité géré personnalisé.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Règles entrantes				
Tous les ICMP-IPv4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré pour l'instance principale. En d'autres termes, le même groupe de sécurité dans lequel la règle s'affiche.	Ces règles réflexives autorisent le trafic entrant à partir de toute instance associée au groupe de sécurité spécifié et atteignable à partir du sous-réseau privé. L'utilisation de la valeur par défaut `ElasticMapReduce-Master-Private` pour plusieurs clusters autorise les nœuds principaux et de tâches de ces clusters à communiquer entre eux sur ICMP ou sur n'importe quel port TCP ou UDP. Spécifiez les groupes de sécurité gérés personnalisés pour restreindre l'accès inter-cluster.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré spécifié pour les nœuds principaux et de tâches.	Ces règles autorisent tout le trafic ICMP entrant et le trafic sur n'importe quel port TCP ou UDP à partir de toutes les instances principales et de tâches qui sont associées au groupe de sécurité spécifié et atteignables à partir du sous-réseau privé, même si les instances se trouvent dans des clusters différents.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
HTTPS (8443)	TCP	8443	L'ID de groupe du groupe de sécurité géré pour l'accès au service dans un sous-réseau privé.	Cette règle permet au gestionnaire de cluster de communiquer avec le nœud principal.

Groupe de sécurité géré par Amazon EMR pour les instances principales et de tâches (sous-réseaux privés)

Le groupe de sécurité géré par défaut pour les instances principales et de tâches dans les sous-réseaux privés possède le Group Name (Nom de groupe) ElasticMapReduce-master. Le groupe de sécurité géré par défaut comporte les règles suivantes et Amazon EMR ajoute les mêmes règles si vous spécifiez un groupe de sécurité géré personnalisé.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Règles entrantes				
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré spécifié pour les instances principales et de tâches. En	Ces règles réflexives autorisent le trafic entrant à partir de toute instance associée au groupe de sécurité spécifié. Utilisation de la valeur par défaut `ElasticMapReduce-slave` pour plusieurs clusters autorise les instances
Tous les TCP	TCP	Toutes		

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Tous les UDP	UDP	Toutes	d'autres termes, le même groupe de sécurité dans lequel la règle s'affiche.	principales et de tâches de ces clusters à communiquer entre elles sur ICMP ou sur n'importe quel port TCP ou UDP. Spécifiez les groupes de sécurité gérés personnalisés pour restreindre l'accès inter-cluster.
Tous les ICMP-IPV4	Toutes	S/O	L'ID de groupe du groupe de sécurité géré pour l'instance principale.	Ces règles autorisent tout le trafic ICMP entrant et le trafic sur n'importe quel port TCP ou UDP à partir de toutes les instances principales qui sont associées au groupe de sécurité spécifié, même si les instances se trouvent dans des clusters différents.
Tous les TCP	TCP	Toutes		
Tous les UDP	UDP	Toutes		
HTTPS (8443)	TCP	8443	L'ID de groupe du groupe de sécurité géré pour l'accès au service dans un sous-réseau privé.	Cette règle permet au gestionnaire de cluster de communiquer avec les nœuds principaux et de tâches.

Groupe de sécurité géré par Amazon EMR pour l'accès au service (sous-réseaux privés)

Le groupe de sécurité géré par défaut pour l'accès au service dans les sous-réseaux privés possède le Group Name (Nom de groupe) ElasticMapReduce-master. Il a des règles entrantes et des règles sortantes qui autorisent le trafic par HTTPS (port 8443, port 9443) aux autres groupes de sécurité gérés dans des sous-réseaux privés. Ces règles permettent au gestionnaire de cluster de communiquer avec le nœud principal, ainsi qu'avec les nœuds de base et de tâche. Les mêmes règles sont nécessaires si vous utilisez des groupes de sécurité personnalisés.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
Règles sortantes Requises pour tous les clusters EMR				
HTTPS (8443)	TCP	8443	L'ID de groupe du groupe de sécurité géré pour l'instance principale.	Ces règles permettent au gestionnaire de cluster de communiquer avec le nœud principal, ainsi qu'avec les nœuds de base et de tâche.
HTTPS (8443)	TCP	8443	L'ID de groupe du groupe de sécurité géré spécifié pour les instances principales et de tâches.	Ces règles permettent au gestionnaire de cluster de communiquer avec le nœud principal, ainsi qu'avec les nœuds de base et de tâche.
Règles entrantes Requises pour les clusters EMR avec Amazon EMR version 5.30.0 et ultérieures.				
HTTPS (9443)	TCP	9443	L'ID de groupe du groupe de sécurité	Cette règle permet la communication entre le groupe de sécurité de l'instance principale et le groupe de sécurité d'accès au service.

Tapez	Protoco	Plage de ports	Source	Details (Détails)
			géré pour l'instance principale.	

Utilisation des groupes de sécurité supplémentaires

Que vous utilisez les groupes de sécurité gérés par défaut ou que vous spécifiez des groupes de sécurité gérés personnalisés, vous pouvez utiliser des groupes de sécurité supplémentaires. Les groupes de sécurité supplémentaires vous donnent la flexibilité nécessaire pour adapter l'accès entre les différents clusters et des clients externes, des ressources et des applications.

Prenez comme exemple le scénario suivant. Vous avez plusieurs clusters que vous voulez faire communiquer entre eux, mais vous voulez autoriser l'accès SSH entrant vers l'instance principale uniquement pour un sous-ensemble particulier de clusters. Pour ce faire, vous pouvez utiliser le même ensemble de groupes de sécurité gérés pour les clusters. Vous pouvez ensuite créer des groupes de sécurité supplémentaires qui autorisent l'accès SSH entrant depuis les clients de confiance et spécifier les groupes de sécurité supplémentaires pour l'instance principale pour chaque cluster du sous-ensemble.

Vous pouvez appliquer jusqu'à quatre groupes de sécurité supplémentaires pour l'instance principale, quatre pour les instances principales et de tâches et quatre pour l'accès au service (dans des sous-réseaux privés). Si nécessaire, vous pouvez spécifier le même groupe de sécurité supplémentaire pour les instances principales, les instances principales et de tâches, et l'accès au service. Le nombre maximal de groupes de sécurité et les règles de votre compte sont soumis à des limites de compte. Pour plus d'informations, consultez Security Group Limits (Limites de groupes de sécurité) dans le Amazon VPC Guide de l'utilisateur.

Spécification de groupes gérés et supplémentaires Amazon EMR

Vous pouvez spécifier les groupes de sécurité à l'aide de la AWS Management Console, de l'AWS CLI ou de l'API EMR. Si vous ne spécifiez pas de groupes de sécurité, Amazon EMR crée des groupes de sécurité par défaut. La spécification de groupes de sécurité supplémentaires est facultative. Vous pouvez attribuer des groupes de sécurité supplémentaires pour les instances principales, les instances principales et de tâches, et l'accès au service (sous-réseaux privés uniquement).

Pour spécifier des groupes de sécurité à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Create cluster (Créer un cluster) et Go to advanced options (Accéder aux options avancées).
3. Choisissez des options pour votre cluster jusqu'à ce que vous atteigniez Étape 4 : Sécurité.
4. Choisissez EC2 Security Groups (Groupes de sécurité EC2) pour développer la section.

   Sous Groupes de sécurité gérés EMR, les groupes de sécurité gérés par défaut sont sélectionnées par défaut. Si aucune valeur par défaut n'existe pas dans le VPC pour Master (Principal), Core & Task (Principal et tâche), ou Accès au service (sous-réseau privé uniquement), Create (Créer) s'affiche avant le nom de groupe de sécurité associé.
5. Si vous utilisez des groupes de sécurité gérés personnalisés, sélectionnez-les à partir des listes Groupes de sécurité gérés EMR.

Si vous sélectionnez un groupe de sécurité géré personnalisé, un message vous informe de sélectionner un groupe de sécurité personnalisé pour d'autres instances. Vous pouvez utiliser uniquement des groupes de sécurité gérés par défaut ou personnalisés pour un cluster.

6. Le cas échéant, sous Additional security groups (Groupes de sécurité supplémentaires), cliquez sur l'icône en forme de crayon, sélectionnez jusqu'à quatre groupes de sécurité dans la liste, puis choisissez Attribuer les groupes de sécurité. Répétez cette opération pour chaque Master (Principal), Core & Task (Principal et tâche), et Service Access (Accès au service), comme vous le souhaitez.
7. Choisissez Créer un cluster.

Spécification des groupes de sécurité à l'aide de l'AWS CLI

Pour spécifier des groupes de sécurité à l'aide de l'AWS CLI, vous utilisez la commande `create-cluster` avec les paramètres suivants de l'option `--ec2-attributes` :

Paramètre	Description
`EmrManagedMasterSecurityGroup`	Utilisez ce paramètre pour spécifier un groupe de sécurité géré personnalisé pour l'instance principale. Si ce paramètre est spécifié, `EmrManagedSlaveSecurityGroup` doit également être spécifié. Pour les clusters dans des sous-réseaux privés, `ServiceAccessSecurityGroup` doit également être spécifié.
`EmrManagedSlaveSecurityGroup`	Utilisez ce paramètre pour spécifier un groupe de sécurité géré personnalisé pour les instances principales et de tâches. Si ce paramètre est spécifié, `EmrManagedMasterSecurityGroup` doit également être spécifié. Pour les clusters dans des sous-réseaux privés, `ServiceAccessSecurityGroup` doit également être spécifié.
`ServiceAccessSecurityGroup`	Utilisez ce paramètre pour spécifier un groupe de sécurité géré personnalisé pour l'accès au service qui s'applique uniquement aux clusters dans des sous-réseaux privés. Le groupe de sécurité que vous spécifiez en tant que `ServiceAccessSecurityGroup` ne doit pas être utilisé à d'autres fins et doit également être réservé à Amazon EMR. Si ce paramètre est spécifié, `EmrManagedMasterSecurityGroup` doit également être spécifié.
`AdditionalMasterSecurityGroups`	Utilisez ce paramètre pour spécifier jusqu'à quatre groupes de sécurité gérés personnalisés pour l'instance principale.
`AdditionalSlaveSecurityGroups`	Utilisez ce paramètre pour spécifier jusqu'à quatre groupes de sécurité gérés personnalisés pour les instances principales et de tâches.

Example — Spécifiez des groupes de sécurité personnalisés gérés par Amazon EMR et des groupes de sécurité supplémentaires

L'exemple suivant spécifie les groupes Amazon EMR de sécurité gérés personnalisés pour un cluster dans un sous-réseau privé, plusieurs groupes de sécurité supplémentaires pour l'instance principale et un seul groupe de sécurité supplémentaire pour les instances principales et de tâches.

> Note
>
> Les caractères de continuité de ligne Linux (\) sont inclus à des fins de lisibilité. Ils peuvent être supprimés ou utilisés dans des commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un accent circonflexe (^).

```
aws emr create-cluster --name "ClusterCustomManagedAndAdditionalSGs" \
--release-label emr-emr-5.30.1 --applications Name=Hue Name=Hive \
Name=Pig --use-default-roles --ec2-attributes \
SubnetIds=subnet-xxxxxxxxxxxx,KeyName=myKey,\
ServiceAccessSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedMasterSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedSlaveSecurityGroup=sg-xxxxxxxxxxxx,\
AdditionalMasterSecurityGroups=['sg-xxxxxxxxxxxx',\
'sg-xxxxxxxxxxxx','sg-xxxxxxxxxxx'],\
AdditionalSlaveSecurityGroups=sg-xxxxxxxxxxxx \
--instance-type m5.xlarge
```

Pour de plus amples informations, veuillez consulter create-cluster dans le AWS CLI Command Reference.

Spécification des groupes de sécurité EC2 pour les blocs-notes EMR

Lorsque vous créez un Bloc-notes EMR, deux groupes de sécurité sont utilisés pour contrôler le trafic réseau entre les blocs-notes Bloc-notes EMR et le cluster Amazon EMR lorsque l'éditeur est utilisé. Les groupes de sécurité par défaut disposent de règles minimales qui autorisent uniquement le trafic réseau entre le service Blocs-notes EMR et les clusters auxquels les blocs-notes sont attachés.

Un Bloc-notes EMR utilise Apache Livy pour communiquer avec le cluster via un proxy à l'aide du port TCP 18888. En créant des groupes de sécurité personnalisés avec des règles adaptées à votre environnement, vous pouvez limiter le trafic réseau afin que seul un sous-ensemble de blocs-notes peut exécuter du code dans l'éditeur de bloc-notes sur des clusters particuliers. Les groupes de sécurité sont utilisés en plus des groupes de sécurité pour le cluster. Pour plus d'informations, consultez Contrôler le trafic de réseau avec les groupes de sécurité dans le Amazon EMR Guide de gestion et dans Spécification des groupes de sécurité EC2 pour les blocs-notes EMR (p. 311).

Groupe de sécurité EC2 par défaut pour l'instance principale

Le groupe de sécurité EC2 par défaut pour l'instance principale est associé à l'instance principale, en plus des groupes de sécurité du cluster pour l'instance principale.

Nom du groupe : Éditeurs de réduction de carte élastique -Livy

Rules

- Entrant

  Autoriser le Port TCP 18888 à partir de n'importe quelle ressource dans le groupe de sécurité EC2 par défaut pour Blocs-notes EMR
- Sortant

  None (Aucun)

Groupe de sécurité EC2 par défaut pour les blocs-notes EMR

Le groupe de sécurité EC2 par défaut pour Bloc-notes EMR est associé à l'éditeur de bloc-notes pour tout Bloc-notes EMR auquel il est attribué.

Nom du groupe : Éditeur-Éditeur ElasticMapReduce

Rules

- Entrant

  None (Aucun)
- Sortant

  Autoriser le Port TCP 18888 à toute ressource dans le groupe de sécurité EC2 par défaut pour Blocs-notes EMR.

Groupe de sécurité EC2 personnalisé pour les bloc-notes EMR lors de l'association de bloc-notes à des référentiels Git

Pour lier un référentiel Git à votre bloc-notes, le groupe de sécurité du bloc-notes EMR doit inclure une règle sortante autorisant le bloc-notes à acheminer le trafic vers Internet. Il est recommandé de créer un nouveau groupe de sécurité à cet effet. La mise à jour du groupe de sécurité ElasticmapReduceEditors-Editor par défaut peut donner les mêmes règles sortantes à d'autres blocs-notes attachés à ce groupe de sécurité.

Rules

- Entrant

  None (Aucun)
- Sortant

  Autorisez le bloc-notes à acheminer le trafic vers Internet via le cluster, comme le montre l'exemple suivant :

Tapez	Protocol	Plage de ports	Destination
Règle TCP personnalisée	TCP	18888	SG-
HTTPS	TCP	443	0.0.0.0/0

Utilisation d'Amazon EMR Block Public Access

Le blocage d'accès public d'Amazon EMR empêche le lancement d'un cluster lorsqu'un groupe de sécurité associé au cluster contient une règle qui autorise le trafic entrant à partir d'une adresse IPv4 0.0.0.0/0 ou IPv6 ::/0 (accès public) sur un port, sauf dans le cas d'un port spécifié en tant qu'exception. Le port 22 est une exception par défaut. Vous pouvez configurer des exceptions pour autoriser l'accès public sur un port ou une plage de ports. En outre, vous pouvez activer ou désactiver le blocage d'accès public. Nous vous recommandons de l'activer.

Le blocage d'accès public est activé et configuré pour chaque région AWS pour votre compte AWS. En d'autres termes, chaque région possède une configuration de blocage d'accès public qui s'applique à tous les clusters créés par votre compte dans cette région.

Note

Pour les comptes qui ont créé des clusters dans une région avant le 25 novembre 2019, le blocage de l'accès public est désactivé par défaut dans cette région. Pour utiliser cette fonction, vous devez l'activer et la configurer manuellement. Pour les comptes qui n'ont pas créé de cluster EMR dans une région avant cette date, le blocage de l'accès public est activé par défaut dans cette région.

Configurer la blocage d'accès public

Vous pouvez activer et désactiver les paramètres de blocage d'accès public à l'aide de la AWS Management Console, l'AWS CLI, de de l'API Amazon EMR. Les paramètres s'appliquent à votre compte région par région.

Pour configurer un blocage d'accès public à l'aide de la AWS Management Console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Dans la barre de navigation, assurez-vous que la Region (Région) que vous souhaitez configurer est sélectionnée.
3. Utilisez Block public access (Blocage d'accès public).
4. Sous Block public access settings (Bloquer les paramètres de blocage d'accès public), suivez les étapes ci-dessous.

Pour...	Faites ceci...
Activer ou désactiver le blocage d'accès public	Choisissez Change (Modifier), choisissez On (Activé) ou Off (Désactivé) le cas échéant, puis cochez la case pour confirmer.
Modifier les ports dans la liste des exceptions	1. Sous Exceptions, choisissez Edit (Modifier). 2. Pour ajouter des ports à la liste des exceptions, choisissez Add a port range (Ajouter une plage de ports) et entrez un nouveau port ou une nouvelle plage de ports. Répétez cette opération pour chaque port ou plage de ports à ajouter. 3. Pour supprimer un port ou une plage de ports, choisissez le x en regard de l'entrée dans la liste Port ranges (Plage de ports) . 4. Choisissez Enregistrer les modifications.

Pour configurer un blocage d'accès public à l'aide de la AWS CLI

Utilisez la commande `aws emr put-block-public-access-configuration` pour configurer le blocage d'accès public comme le montre les exemples suivants.

Pour...	Faites ceci...
Activer le blocage d'accès public	Définissez `BlockPublicSecurityGroupRules` sur `true` comme le montre l'exemple suivant. Pour que le cluster se lance, aucun groupe de sécurité associé à un cluster ne peut avoir une règle entrante qui autorise l'accès public.

Pour...	Faites ceci...
	`aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=true`
Désactiver le blocage d'accès public	Définissez `BlockPublicSecurityGroupRules` sur `false` comme le montre l'exemple suivant. Les groupes de sécurité associés à un cluster peuvent avoir des règles entrantes qui autorisent l'accès public sur n'importe quel port. Nous déconseillons cette configuration. `aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=false`
Activez le blocage d'accès public et spécifiez les ports en tant qu'exceptions	L'exemple suivant active le blocage d'accès public et spécifie le port 22 et les ports 100-101 comme exceptions. Cela permet de créer des clusters si un groupe de sécurité associé possède une règle entrante qui autorise l'accès public sur le port 22, le port 100 ou le port 101. `aws emr put-block-public-access-configuration --block-public-access-configuration '{ "BlockPublicSecurityGroupRules": true, "PermittedPublicSecurityGroupRuleRanges": [ { "MinRange": 22, "MaxRange": 22 }, { "MinRange": 100, "MaxRange": 101 } ] }'`

Validation de la conformité pour Amazon EMR

Les auditeurs tiers évaluent la sécurité et la conformité de Amazon EMR dans le cadre de plusieurs programmes de conformité AWS. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et autres.

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.

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour plus d'informations, consultez Téléchargement des rapports dans AWS Artifact.

Votre responsabilité de conformité lors de l'utilisation de Amazon EMR est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. Si votre utilisation de Amazon EMR est soumise à la conformité aux normes HIPAA, PCI, ou FedRAMP, AWS fournit des ressources pour vous aider :

- Guides de démarrage rapide de la sécurité et de la conformité – Ces guides de déploiement proposent des considérations architecturales et fournissent des étapes pour déployer des environnements de référence centrés sur la sécurité et la conformité sur AWS.

- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA – Le livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications conformes à la loi HIPAA.
- Ressources de conformité AWS – Cet ensemble de manuels et de guides peut s'appliquer à votre secteur et à votre emplacement.
- AWS Config – Ce service AWS permet d'évaluer comment les configurations de vos ressources se conforment aux pratiques internes, aux normes et aux directives industrielles.
- AWS Security Hub – Ce service AWS fournit une vue complète de votre état de sécurité au sein AWS qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.

Résilience dans Amazon EMR

L'infrastructure mondiale dAWS repose sur des régions et des zones de disponibilité AWS. Les régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité 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 plus d'informations sur les régions et les zones de disponibilité AWS, consultez Infrastructure mondiale AWS.

Outre l'infrastructure globale AWS, Amazon EMR propose plusieurs fonctionnalités qui contribuent à la prise en charge des vos besoins en matière de résilience et de sauvegarde de données.

- Intégration à Amazon S3 via EMRFS
- Prise en charge de plusieurs nœuds principaux

Sécurité de l'infrastructure dans Amazon EMR

En tant que service géré, Amazon EMR est protégé par les procédures de sécurité du réseau mondial AWS qui sont décrites dans le livre blanc Amazon Web Services : Présentation des procédures de sécurité.

Vous utilisez les appels d'API publiés AWS pour accéder à Amazon EMR via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.0 ou version ultérieure. Nous recommandons TLS 1.2 ou version ultérieure. Les clients doivent également 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 telles que Java 7 et 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 mandataire IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Rubriques

- Connexion à Amazon EMR à l'aide d'un point de terminaison de VPC d'interface (p. 315)

Connexion à Amazon EMR à l'aide d'un point de terminaison de VPC d'interface

Vous pouvez vous connecter directement à Amazon EMR à l'aide d'un point de terminaison de VPC d'interface (AWS PrivateLink) dans votre Virtual Private Cloud (VPC) au lieu de vous connecter via Internet.

Lorsque vous utilisez un point de terminaison de VPC d'interface, la communication entre votre VPC et Amazon EMR est gérée entièrement au sein du réseau AWS. Chaque point de terminaison d'un VPC est représenté par une ou plusieurs interfaces réseau Elastic (ENI) avec des adresses IP privées dans vos sous-réseaux VPC.

Le point de terminaison de VPC d'interface connecte votre VPC directement à Amazon EMR sans passerelle Internet, périphérique NAT, connexion VPN ni connexion AWS Direct Connect. Les instances de votre VPC ne nécessitent pas d'adresses IP publiques pour communiquer avec l'API Amazon EMR.

Pour utiliser Amazon EMR via votre VPC, vous devez vous connecter à partir d'une instance située dans le VPC ou connecter votre réseau privé à votre VPC à l'aide d'un Amazon Virtual Private Network (VPN) ou d'AWS Direct Connect. Pour obtenir des informations sur Amazon VPN, consultez Connexions VPN dans le guide de l'utilisateur Amazon Virtual Private Cloud. Pour obtenir des informations sur AWS Direct Connect, consultez Création d'une connexion dans le guide de l'utilisateur AWS Direct Connect.

Vous pouvez créer un point de terminaison de VPC d'interface pour vous connecter à Amazon EMR à l'aide de la console AWS ou des commandes AWS Command Line Interface (AWS CLI). Pour de plus amples informations, veuillez consulter Création d'un point de terminaison d'interface.

Une fois que vous avez créé un point de terminaison de VPC d'interface, si vous activez les noms d'hôte DNS privés pour le point de terminaison, le point de terminaison Amazon EMR par défaut est résolu par votre point de terminaison de VPC. Le point de terminaison par défaut du nom de service Amazon EMR a le format suivant.

```
elasticmapreduce.Region.amazonaws.com
```

Si vous n'activez pas les noms d'hôte DNS privés, Amazon VPC fournit un nom de point de terminaison DNS que vous pouvez utiliser au format suivant.

```
VPC_Endpoint_ID.elasticmapreduce.Region.vpce.amazonaws.com
```

Pour de plus amples informations, veuillez consulter Points de terminaison de VPC d'interface (AWS PrivateLink) dans le Amazon VPC Guide de l'utilisateur.

Amazon EMR prend en charge l'exécution d'appels en direction de toutes ses actions d'API à l'intérieur de votre VPC.

Vous pouvez attacher des stratégies de point de terminaison VPC à un point de terminaison VPC pour contrôler l'accès des entités IAM. Vous pouvez également associer des groupes de sécurité à un point de terminaison VPC pour contrôler l'accès entrant et sortant en fonction de l'origine et de la destination du trafic réseau, comme une plage d'adresses IP. Pour plus d'informations, consultez Contrôle de l'accès aux services avec des points de terminaison de VPC.

Créer une stratégie de point de terminaison VPC pour Amazon EMR

Vous pouvez créer une stratégie pour les points de terminaison Amazon VPC pour Amazon EMR dans laquelle vous pouvez spécifier :

- Principal qui peut ou ne peut pas effectuer des actions
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour de plus amples informations, consultez Contrôle de l'accès aux services avec des points de terminaison de VPC dans le Guide de l'utilisateur Amazon VPC.

Example Stratégie de point de terminaison VPC – pour refuser tout accès à partir d'un compte AWS spécifié

La stratégie de point de terminaison VPC suivante refuse au compte AWS *123456789012* tout accès aux ressources utilisant le point de terminaison.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example Stratégie de point de terminaison VPC – pour autoriser l'accès VPC uniquement à un principal IAM spécifié (utilisateur)

La stratégie de point de terminaison VPC suivante permet un accès complet uniquement à l'utilisateur IAM *lijuan* dans le compte AWS *123456789012*. Toutes les autres entités IAM se voient refuser l'accès à l'aide du point de terminaison.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
                ]
            }
        }]
}
```

Example Stratégie de point de terminaison VPC – pour autoriser les opérations EMR en lecture seule

La stratégie de point de terminaison VPC suivante autorise uniquement le compte AWS *123456789012* à effectuer les actions Amazon EMR spécifiées.

Les actions spécifiées fournissent l'équivalent d'un accès en lecture seule pour Amazon EMR. Toutes les autres actions sur le VPC sont refusées pour le compte spécifié. Tous les autres comptes se voient refuser tout accès. Pour afficher une liste d'actions Amazon EMR, veuillez consulter Actions, ressources et clés de condition pour Amazon EMR.

```
{
```

```
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example Stratégie de point de terminaison VPC – refusant l'accès à un cluster spécifié

La stratégie de point de terminaison VPC suivante permet un accès complet à tous les comptes et entités, mais refuse tout accès pour le compte AWS *123456789012* aux actions effectuées sur le cluster Amazon EMR avec l'ID de cluster *J-A1B2CD34ef5g*. D'autres actions Amazon EMR qui ne prennent pas en charge les autorisations au niveau des ressources pour les clusters sont toujours autorisées. Pour obtenir la liste des actions Amazon EMR et leur type de ressource correspondant, veuillez consulter Actions, Ressources et Clés de condition pour Amazon EMR.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-west-2:123456789012:cluster/j-
A1B2CD34EF5G",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Manage Clusters

Une fois que vous avez lancé votre cluster, vous pouvez le surveiller et le gérer. Amazon EMR fournit plusieurs outils que vous pouvez utiliser pour vous connecter à votre cluster et le contrôler.

Rubriques

- View and Monitor a Cluster (p. 319)
- Connexion au cluster (p. 367)
- Arrêter un cluster (p. 382)
- Dimensionnement des ressources de cluster (p. 384)
- Cloning a Cluster Using the Console (p. 417)
- Submit Work to a Cluster (p. 418)
- Automate Recurring Clusters with AWS Data Pipeline (p. 425)

View and Monitor a Cluster

Amazon EMR fournit plusieurs outils que vous pouvez utiliser pour obtenir des informations sur votre cluster. Vous pouvez accéder aux informations sur le cluster à partir de la console, de l'interface de ligne de commande ou par programmation. Les interfaces web Hadoop standard et les fichiers journaux sont disponibles sur le nœud maître. Vous pouvez également utiliser les services de surveillance comme CloudWatch et Ganglia pour suivre les performances de votre cluster.

L'historique d'application est également disponible à partir de la console via les interfaces utilisateur d'application « persistante » pour le serveur d'historique Spark à partir d'Amazon EMR 5.25.0. Avec Amazon EMR 6.x, le serveur de chronologie YARN persistant et les interfaces utilisateur Tez sont également disponibles. Ces services sont hébergés en dehors du cluster, de sorte que vous pouvez accéder à l'historique d'application pendant 30 jours après la mise hors service du cluster sans avoir besoin d'une connexion SSH ou d'un proxy Web. Veuillez consulter Afficher l'historique d'application.

Rubriques

- View Cluster Status and Details (p. 319)
- Enhanced Step Debugging (p. 325)
- Afficher l'historique de l'application (p. 327)
- View Log Files (p. 333)
- View Cluster Instances in Amazon EC2 (p. 337)
- Événements et métriques CloudWatch (p. 338)
- View Cluster Application Metrics with Ganglia (p. 364)
- Journalisation des appels d'API Amazon EMR dans AWS CloudTrail (p. 365)

View Cluster Status and Details

Une fois que vous avez créé un cluster, vous pouvez surveiller son statut et obtenir des informations détaillées sur son exécution et sur les erreurs qui ont pu se produire, même après que l'exécution a pris

fin. Amazon EMR enregistre pendant deux mois les métadonnées sur des clusters résiliés pour référence, délai après lequel les métadonnées sont supprimées. Vous ne pouvez pas supprimer de clusters à partir de l'historique des clusters, mais à l'aide de la AWS Management Console, vous pouvez utiliser la commande Filtrer, tandis qu'avec l'AWS CLI, vous pouvez utiliser les options de la commande `list-clusters` pour vous concentrer sur les clusters dont vous vous occupez.

Vous pouvez accéder à l'historique d'application stocké sur le cluster pendant une semaine à compter de la date de son enregistrement, que le cluster soit en cours d'exécution ou hors service. En outre, les interfaces utilisateur d'application persistante stockent l'historique d'application hors cluster pendant 30 jours après la mise hors service d'un cluster. Veuillez consulter Afficher l'historique d'application.

View Cluster Status Using the AWS Management Console

La Clusters List (Liste de clusters) de la console Amazon EMR affiche tous les clusters de votre compte et de votre région AWS, y compris les clusters résiliés. La liste affiche les informations suivantes pour chaque cluster : le nom et l'ID, le statut, la date de création du cluster, le temps écoulé depuis le début d'exécution du cluster et les heures d'instances normalisées accumulées pour toutes les instances EC2 du cluster. Cette liste est le point de départ pour surveiller le statut de vos clusters. Elle est conçue de sorte que vous puissiez explorer les détails de chaque cluster à des fins d'analyse et de dépannage.

Pour afficher un résumé abrégé des informations du cluster

- Sélectionnez la flèche vers le bas à côté du lien du cluster sous Nom.

  La ligne du cluster se développe pour afficher plus d'informations sur le cluster, le matériel, les étapes et les actions d'amorçage. Utilisez les liens de cette section pour explorer les détails spécifiques. Par exemple, cliquez sur un lien sous Étapes pour accéder aux fichiers journaux, voir le JAR associé à l'étape, explorer les travaux et les tâches de l'étape et accéder aux fichiers journaux.

Pour afficher le statut détaillé d'un cluster

- Sélectionnez le lien du cluster sous Nom pour ouvrir la page des détails du cluster. Utilisez chaque onglet pour afficher les informations comme décrit dans la section suivante.

  Utilisez chaque onglet pour les informations suivantes :

Onglet	Informations
Résumé	Utilisez cet onglet pour afficher les informations de base de la configuration du cluster, comme l'URL à utiliser pour les connexions SSH au nœud principal, les applications open source installées par Amazon EMR lors de la création du cluster, l'emplacement de stockage des journaux dans Amazon S3 et la version Amazon EMR utilisée pour créer le cluster.
Application user interfaces (Interfaces utilisateur d'application)	Utilisez cet onglet pour afficher les détails d'application persistante du serveur de chronologie YARN et de l'interface utilisateur Tez hors cluster. Pour les travaux Spark, vous pouvez explorer en détail les informations disponibles sur les travaux, les phases et les exécuteurs. Pour plus d'informations, consultez Afficher l'historique de l'application (p. 327), Les interfaces utilisateur d'application sur cluster sont disponibles pendant l'exécution du cluster.
: Monitoring	Utilisez cet onglet pour afficher les graphiques décrivant les indicateurs clés du fonctionnement d'un cluster sur une période de temps que vous spécifiez. Vous pouvez afficher les données du niveau cluster, les données du niveau nœud et les informations sur les E/S et le stockage des données.

Onglet	Informations
Matériel	Utilisez cet onglet pour afficher les informations sur les nœuds de votre cluster, y compris les ID des instances EC2, les noms DNS, les adresses IP et plus encore.
Événements .	Utilisez cet onglet pour afficher le journal des événements de votre cluster. Pour plus d'informations, consultez Supervision des événements CloudWatch (p. 339),
Étapes	Utilisez cet onglet pour afficher fichiers journaux du statut et de l'accès des étapes que vous avez soumises. Pour plus d'informations sur ces étapes, consultez Work with Steps Using the AWS CLI and Console (p. 418).
Configurations	Utilisez cet onglet pour afficher tous les objets de configuration personnalisée appliqués au cluster. Pour en savoir plus sur les classifications de configurations, consultez Configuration des applications dans le Amazon EMR Guide de version.
Actions d'amorçage	Utilisez cet onglet pour afficher le statu des actions d'amorçage que le cluster exécute lors de son lancement. Les actions d'amorçage sont utilisées pour les installations logicielles personnalisées et les configurations avancées. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125),

View Cluster Status Using the AWS CLI

Les exemples suivants illustrent comment récupérer les détails du cluster à l'aide de l'AWS CLI. Pour plus d'informations sur les commandes disponibles, consultez la Référence des commandes de l'AWS CLI pour Amazon EMR. Vous pouvez utiliser la commande describe-cluster pour afficher les détails au niveau du cluster, y compris le statut, la configuration matérielle et logicielle, les paramètres VPC, les actions d'amorçage, les groupes d'instances, etc. L'exemple suivant illustre l'utilisation de la commande `describe-cluster`, suivie par des exemples de la commande list-clusters.

Example Affichage du statut du cluster

Pour utiliser la commande `describe-cluster`, vous avez besoin de l'ID de cluster. Cet exemple illustre comment obtenir une liste de clusters créés en une plage de temps donnée, ainsi que l'utilisation de l'un des ID de cluster retournés pour afficher plus d'informations sur le statut d'un cluster.

La commande suivante décrit le cluster *`j-1K48XXXXXXHCB`*, que vous remplacez par votre ID de cluster.

```
aws emr describe-cluster --cluster-id j-1K48XXXXXXHCB
```

Le sortie de votre commande est semblable à l'exemple suivant :

```
{
```

```
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1438281058.061,
                "CreationDateTime": 1438280702.498
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "EmrManagedMasterSecurityGroup": "sg-cXXXXX0",
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2KeyName": "myKey",
            "Ec2AvailabilityZone": "us-east-1c",
            "EmrManagedSlaveSecurityGroup": "sg-example"
        },
        "Name": "Development Cluster",
        "ServiceRole": "EMR_DefaultRole",
        "Tags": [],
        "TerminationProtected": false,
        "ReleaseLabel": "emr-4.0.0",
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1438281058.101,
                        "CreationDateTime": 1438280702.499
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "CORE",
                "InstanceGroupType": "CORE",
                "Id": "ig-2EEXAMPLEXXP",
                "Configurations": [],
                "InstanceType": "m5.xlarge",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1438281023.879,
                        "CreationDateTime": 1438280702.499
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "MASTER",
                "InstanceGroupType": "MASTER",
                "Id": "ig-2A1234567XP",
                "Configurations": [],
                "InstanceType": "m5.xlarge",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            }
        ],
```

```
        "Applications": [
            {
                "Version": "1.0.0",
                "Name": "Hive"
            },
            {
                "Version": "2.6.0",
                "Name": "Hadoop"
            },
            {
                "Version": "0.14.0",
                "Name": "Pig"
            },
            {
                "Version": "1.4.1",
                "Name": "Spark"
            }
        ],
        "BootstrapActions": [],
        "MasterPublicDnsName": "ec2-X-X-X-X.compute-1.amazonaws.com",
        "AutoTerminate": false,
        "Id": "j-jobFlowID",
        "Configurations": [
            {
                "Properties": {
                    "hadoop.security.groups.cache.secs": "250"
                },
                "Classification": "core-site"
            },
            {
                "Properties": {
                    "mapreduce.tasktracker.reduce.tasks.maximum": "5",
                    "mapred.tasktracker.map.tasks.maximum": "2",
                    "mapreduce.map.sort.spill.percent": "90"
                },
                "Classification": "mapred-site"
            },
            {
                "Properties": {
                    "hive.join.emit.interval": "1000",
                    "hive.merge.mapfiles": "true"
                },
                "Classification": "hive-site"
            }
        ]
    }
}
```

Example Affichage des clusters par date de création

Pour obtenir les clusters créés en une plage de temps donnée, utilisez la commande `list-clusters` avec les paramètres `--created-after` et `--created-before`.

La commande suivante répertorie tous les clusters créés entre le 09/10/2019 et le 12/10/2019.

```
aws emr list-clusters --created-after 2019-10-09T00:12:00 --created-
before 2019-10-12T00:12:00
```

Example Affichage des clusters par état

Pour afficher les clusters par état, utilisez la commande `list-clusters` avec le paramètre `--cluster-states`. Les états de cluster valides comprennent : DÉMARRAGE, AMORÇAGE, RUNNING, WAITING, TERMINATE, TERMINATED, et TERMINATED_WITH_ERRORS.

```
aws emr list-clusters --cluster-states TERMINATED
```

Vous pouvez également utiliser les paramètres de raccourci suivants pour afficher tous les clusters ayant les états spécifiés :

- `--active` filters clusters in the STARTING,BOOTSTRAPPING, RUNNING, WAITING, or TERMINATING states.
- `--terminated` filters clusters in the TERMINATED state.
- `--failed` parameter filters clusters in the TERMINATED_WITH_ERRORS state.

Les commandes suivantes retournent le même résultat.

```
aws emr list-clusters --cluster-states TERMINATED
```

```
aws emr list-clusters --terminated
```

Enhanced Step Debugging

Si une étape Amazon EMR échoue et que vous avez envoyé votre travail à l'aide de l'opération d'API Step avec une AMI de version 5.x ou ultérieure, Amazon EMR peut, dans certains cas, identifier et déterminer la cause de l'échec de l'étape. Il renvoie alors le nom du fichier journal concerné et une partie de la trace de la pile d'application via une API. Les échecs suivants peuvent par exemple être identifiés :

- A common Hadoop error such as the output directory already exists, the input directory does not exist, or an application runs out of memory.
- Java errors such as an application that was compiled with an incompatible version of Java or run with a main class that is not found.
- An issue accessing objects stored in Amazon S3.

Ces informations sont disponibles à l'aide des opérations d'API DescribeStep et ListSteps. Le champ FailureDetails de StepSummary renvoyé par ces opérations. Pour accéder aux informations FailureDetails, utilisez l'interface de ligne de commande AWS, la console ou AWS SDK.

Pour afficher les informations relatives aux échecs à l'aide de la console AWS

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisir Cluster List et sélectionnez un cluster.
3. Sélectionnez l'icône en forme de flèche en regard de chaque étape pour afficher plus d'informations.

Lorsqu'une étape échoue et qu'Amazon EMR peut identifier la cause première, les détails de l'échec s'affichent.

Pour afficher les détails relatifs à un échec à l'aide de l'interface de ligne de commande AWS

- Pour obtenir les détails de l'échec d'une étape à l'aide de l'interface de ligne de commande AWS, utilisez la commande describe-step.

```
aws emr describe-step -cluster-id j-1K48XXXXXHCB -step-id s-3QM0XXXXXM1W
```

La sortie sera similaire à l'exemple suivant.

```
{
  "Step": {
    "Status": {
      "FailureDetails": {
        "LogFile": "s3://myBucket/logs/j-1K48XXXXXHCB/steps/s-3QM0XXXXXM1W/stderr.gz",
        "Message": "org.apache.hadoop.mapred.FileAlreadyExistsException: Output
 directory s3://myBucket/logs/beta already exists",
        "Reason": "Output directory already exists."
      },
      "Timeline": {
        "EndDateTime": 1469034209.143,
        "CreationDateTime": 1469033847.105,
        "StartDateTime": 1469034202.881
      },
      "State": "FAILED",
      "StateChangeReason": {}
    },
    "Config": {
      "Args": [
        "wordcount",
        "s3://myBucket/input/input.txt",
        "s3://myBucket/logs/beta"
      ],
      "Jar": "s3://myBucket/jars/hadoop-mapreduce-examples-2.7.2-amzn-1.jar",
      "Properties": {}
    },
    "Id": "s-3QM0XXXXXM1W",
    "ActionOnFailure": "CONTINUE",
```

```
    "Name": "ExampleJob"
  }
}
```

Afficher l'historique de l'application

Vous pouvez afficher les détails des applications de serveur d'historique Spark et de serveur de chronologie YARN à l'aide de l'onglet Application user interfaces (Interfaces utilisateur d'application) de la page des détails d'un cluster dans la console. L'historique d'application Amazon EMR facilite le dépannage et l'analyse des tâches actives et de l'historique des tâches.

L'onglet Application user interfaces (Interfaces utilisateur d'application) propose plusieurs options d'affichage :

- Accès hors cluster aux interfaces utilisateur d'application persistante – À partir d'Amazon EMR version 5.25.0, des liens d'interface utilisateur d'application persistante sont disponibles pour Spark. Avec Amazon EMR version 6.0.0 et ultérieure, l'interface utilisateur Tez et le serveur de chronologie YARN ont également des interfaces utilisateur d'application persistante. Le serveur de chronologie YARN et l'interface utilisateur Tez sont des applications open source qui fournissent des métriques pour les clusters actifs et hors service. L'interface utilisateur Spark fournit des détails sur les étapes et les tâches du planificateur, les tailles de RDD et l'utilisation de la mémoire, des informations sur l'environnement et des informations sur les programmes d'exécution en cours. Les interfaces utilisateur d'application persistante sont exécutées hors cluster, de sorte que les informations et les journaux de cluster sont disponibles pendant 30 jours après la mise hors service d'une application. Contrairement aux interfaces utilisateur d'application en cluster, les interfaces utilisateur d'application persistante ne nécessitent pas de configurer un proxy Web via une connexion SSH.
- Interfaces utilisateur d'application en cluster – Il existe une variété d'interfaces utilisateur d'historique d'application qui peuvent être exécutées sur un cluster. Les interfaces utilisateur en cluster sont hébergées sur le nœud principal et nécessitent que vous configuriez une connexion SSH au serveur Web. Les interfaces utilisateur d'application en cluster conservent l'historique d'application pendant une semaine après la mise hors service d'une application.
- Historique de l'application de niveau général – Avec Amazon EMR version 5.8.0 ou ultérieure, vous pouvez afficher un récapitulatif de l'historique de l'application dans la console EMR, notamment des métriques clés pour les tâches et les programmes d'exécution d'étape. Le récapitulatif de l'historique de l'application est disponible pour toutes les applications YARN. Des détails supplémentaires sont fournis pour les applications Spark, mais ces détails ne constituent qu'un sous-ensemble des informations disponibles via l'interface utilisateur d'application Spark.

Afficher les interfaces utilisateur d'application persistante

À partir d'Amazon EMR version 5.25.0, vous pouvez vous connecter aux détails de l'application persistante de serveur d'historique Spark hébergée hors cluster à l'aide de la page Summary (Récapitulatif) du cluster ou de l'onglet Application user interfaces (Interfaces utilisateur d'application) de la console. Les interfaces d'application persistantes du serveur de chronologie Tez UI et YARN sont disponibles à partir de Amazon EMR version 5.30.1. L'accès en un clic à un lien vers l'historique d'application persistante offre les avantages suivants :

- Vous pouvez rapidement analyser et dépanner les tâches actives et l'historique des tâches sans configurer de proxy Web via une connexion SSH.
- Vous pouvez accéder à l'historique de l'application et aux fichiers journaux pertinents pour les clusters actifs et hors service. Les journaux sont disponibles pendant 30 jours après la mise hors service de l'application.

Sur la page Interfaces utilisateur d'application ou le cluster Résumé pour votre cluster dans le Amazon EMR 5.30.1 ou console 6.x, choisissez le serveur de chronologie YARN, Interface utilisateur de Tezou Serveur d'historique Spark lien.

L'interface utilisateur de l'application s'ouvre dans un nouvel onglet de navigateur. Pour plus d'informations, consultez les informations sur la surveillance et l'instrumentation.

Vous pouvez afficher les journaux de conteneur YARN via les liens sur le serveur d'historique Spark, le serveur de chronologie YARN et l'interface utilisateur Tez.

> **Note**
>
> Pour accéder aux journaux du conteneur YARN à partir du serveur d'historique Spark, du serveur de chronologie YARN et de l'interface utilisateur Tez, vous devez activer la journalisation sur Amazon S3 pour votre cluster. Si la journalisation n'est pas activée, les liens vers les journaux de conteneur YARN ne fonctionneront pas.

Collecte des journaux

Pour activer l'accès en un clic aux interfaces utilisateur d'application persistante, Amazon EMR collecte deux types de journaux :

- Lesjournaux d'événements d'application, collectés dans un compartiment système EMR. Les journaux d'événements sont chiffrés au repos à l'aide du chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3). Si vous utilisez un sous-réseau privé pour votre cluster, assurez-vous d'inclure `"arn:aws:s3:::prod.MyRegion.appinfo.src/*"` dans la liste des ressources de la stratégie Amazon S3 pour le sous-réseau privé. Pour plus d'informations, consultez Stratégie Amazon S3 minimum pour un sous-réseau privé.
- Lesjournaux de conteneur YARN sont collectés dans un compartiment Amazon S3 que vous possédez. Vous devez activer la journalisation pour votre cluster pour accéder aux journaux de conteneur YARN. Pour plus d'informations, voir Configuration de la journalisation et du débogage de cluster.

Si vous devez désactiver cette fonctionnalité pour des raisons de confidentialité, vous pouvez arrêter le démon à l'aide d'un script d'amorçage lorsque vous créez un cluster, comme l'illustre l'exemple suivant.

```
aws emr create-cluster --name "Stop Application UI Support" --release-label emr-5.30.1
--applications Name=Hadoop Name=Spark --ec2-attributes KeyName=keyname
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=CORE,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=TASK,InstanceCount=1,InstanceType=m3.xlarge
--use-default-roles --bootstrap-actions Path=s3://elasticmapreduce/bootstrap-actions/run-
if,Args=["instance.isMaster=true","echo Stop Application UI | sudo tee /etc/apppusher/run-
apppusher; sudo systemctl stop apppusher || exit 0"]
```

Après l'exécution de ce script d'amorçage, Amazon EMR ne collectera pas de journaux d'événements de serveur d'historique Spark ou de serveur de chronologie YARN dans le compartiment système EMR. Aucune information sur l'historique d'application ne sera disponible dans l'onglet Application user interfaces

(Interfaces utilisateur d'application) et vous perdrez l'accès à toutes les interfaces utilisateur d'application à partir de la console.

Considérations et restrictions

L'accès en un clic aux interfaces utilisateur d'application persistante présente actuellement les limitations suivantes :

- Il y aura un délai d'au moins deux minutes lorsque les détails de l'application apparaîtront sur l'interface utilisateur du serveur d'historique Spark.
- Cette fonction est opérationnelle uniquement lorsque le répertoire de journal d'événements pour l'application est dans le système de fichiers HDFS. Par défaut, Amazon EMR stocke les journaux d'événements dans un répertoire du système de fichiers HDFS. Si vous modifiez le répertoire par défaut en un système de fichiers différent, par exemple Amazon S3, cette fonction ne fonctionnera pas.
- Cette fonction n'est actuellement pas disponible pour les clusters EMR avec plusieurs nœuds principaux ou pour les clusters EMR intégrés à AWS Lake Formation.
- Pour activer l'accès en un clic aux interfaces utilisateur d'application persistante, vous devez disposer des autorisations requises sur l'action `DescribeCluster` pour EMR. Si vous refusez d'accorder l'autorisation nécessaire à l'exécution de cette action à un mandataire IAM, la propagation de la modification d'autorisation prend environ cinq minutes.
- Si vous reconfigurez des applications dans un cluster en cours d'exécution, l'historique de l'application ne sera pas disponible via l'interface utilisateur de l'application.
- Pour chaque compte AWS, le nombre d'interfaces utilisateurs d'application active ne peut pas dépasser 50.
- Vous pouvez accéder aux interfaces utilisateur d'application à partir de la console dans les régions USA Est (Virginie du Nord et Ohio), USA Ouest (Californie du Nord et Oregon), Canada (Centre), UE (Francfort, Irlande, Londres et Paris), Asie-Pacifique (Mumbai, Osaka, Séoul, Singapour, Sydney et Tokyo) et Amérique du Sud (Sao Paulo).

Afficher les interfaces utilisateur d'application en cluster

Les interfaces utilisateur d'application en cluster sont hébergées sur le nœud principal et nécessitent un tunnel SSH vers le serveur Web local du nœud principal. Pour plus d'informations, consultez la section Connexion au nœud maître à l'aide de SSH (p. 368). À l'exception du serveur d'historique Spark, du serveur de chronologie YARN et des applications Hive, l'historique d'application sur cluster ne peut être affiché que pendant l'exécution du cluster. Pour de plus amples informations sur l'affichage des interfaces Web, veuillez consulter Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

Dans l'onglet Application user interfaces (Interfaces utilisateur d'application) ou la page Summary (Récapitulatif) de votre cluster dans la console Amazon EMR 6.x, choisissez le lien Application user interfaces (Interfaces utilisateur d'application) que vous souhaitez afficher.

Affichage d’un récapitulatif de l'historique de l’application

Avec Amazon EMR version 5.8.0 ou ultérieure, vous pouvez afficher un récapitulatif général de l'historique de l’application à partir de l'onglet Application user interfaces (Interfaces utilisateur d'application) dans la console Amazon EMR. Amazon EMR conserve le récapitulatif de l'historique de l’application pendant sept jours après que celle-ci a pris fin.

La séquence suivante illustre l’exploration des détails des tâches d’une application Spark ou YARN via l'Application user interfaces (Interfaces utilisateur d'application) sur la page de détails d'un cluster. Pour afficher les détails du cluster, à partir de l’ Grappes liste, sélectionnez un cluster Nom. Pour afficher des informations sur les journaux de conteneur YARN, vous devez activer la journalisation pour votre cluster. Pour plus d'informations, voir Configuration de la journalisation et du débogage de cluster. Pour l'historique de l’application Spark, les informations fournies dans le tableau récapitulatif ne sont qu'un sous-ensemble des informations disponibles via l'interface utilisateur du serveur d'historique Spark.

Dans l'exemple d'onglet Application user interfaces (Interfaces utilisateur d'application) suivant, une ligne est développée pour afficher le récapitulatif du diagnostic d'une application Spark, et un lien Application ID (ID d’application) est sélectionné pour afficher les détails d'une autre application.

Dans l'exemple ci-dessous, sur l'onglet Jobs (Tâches) de YARN application (Application YARN), la description de Job 9 est sélectionnée et affiche les détails de Job 9.

Sur la page des détails de Job 9, des informations sur les étapes de la tâche sont développées, puis la description de l'étape 29 est sélectionnée de manière à afficher les détails de l'étape 29.

Sur la page de détails de l' étape 29, les métriques clés des tâches et des programmes d'exécution de l'étape sont visibles, et les journaux des tâches et des programmes d'exécution sont accessibles par l'intermédiaire de liens.

View Log Files

Amazon EMR et Hadoop génèrent des fichiers journaux qui indiquent le statut du cluster. Par défaut, ces journaux sont écrits sur le nœud maître dans le répertoire /mnt/var/log/. En fonction de la façon dont vous avez configuré votre cluster lorsque vous l'avez lancé, ces journaux peuvent également être archivés sur Amazon S3 et être affichés grâce à l'outil de débogage graphique.

Il existe de nombreux types de journaux écrits sur le nœud principal. Amazon EMR écrit les journaux d'étape, d'action d'amorçage et d'état d'instances. Apache Hadoop écrit des journaux pour indiquer le traitement des travaux, des tâches et des tentatives de tâche. Hadoop enregistre également les journaux de ses démons. Pour plus d'informations sur les journaux écrits par Hadoop, consultez http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html.

Rubriques

- View Log Files on the Master Node (p. 333)
- View Log Files Archived to Amazon S3 (p. 335)
- View Log Files in the Debugging Tool (p. 336)

View Log Files on the Master Node

Le tableau suivant répertorie quelques uns des fichiers journaux que vous trouverez sur le nœud maître.

Emplacement	Description :
/mnt/var/log/bootstrap-actions	Journaux écrits pendant le traitement des actions amorçage.

Emplacement	Description :
/mnt/var/log/hadoop-state-pusher	Journaux écrits par le processus de transmission d'état Hadoop.
/mnt/var/log/instance-controller (Amazon EMR 4.6.0 et version antérieure) /emr/instance-controller (Amazon EMR 4.7.0 et version ultérieure)	Journaux de contrôleur d'instance.
/mnt/var/log/instance-state	Journaux d'état de l'instance. Ces journaux contiennent des informations sur l'UC, l'état de la mémoire et les threads de nettoyage de mémoire du nœud.
/mnt/var/log/service-nanny (Amazon EMR 4.6.0 et version antérieure) /emr/service-nanny (Amazon EMR 4.7.0 et version ultérieure)	Journaux écrits par le processus de surveillance du service.
/mnt/var/log/*application*	Journaux spécifiques à une application, par exemple Hadoop, Spark ou Hive.
/mnt/var/log/hadoop/steps/*N*	Journaux d'étape qui contiennent des informations sur le traitement de l'étape. La valeur de *N* indique le stéthoscope attribué par Amazon EMR. Par exemple, un cluster comporte deux étapes : `s-1234ABCDEFGH` et `s-5678IJKLMNOP`. La première étape est située dans `/mnt/var/log/hadoop/steps/s-1234ABCDEFGH/` et la deuxième étape `/mnt/var/log/hadoop/steps/s-5678IJKLMNOP/`. Les journaux d'étape suivants sont écrits par Amazon EMR. • contrôleur — Informations sur le traitement de l'étape. Si l'étape échoue lors de son chargement, vous pouvez trouver la trace de la pile dans ce journal. • syslog — Décrit l'exécution des tâches Hadoop dans l'étape. • stderr — Canal d'erreur standard de Hadoop lors du traitement de l'étape. • stdout — Canal de sortie standard de Hadoop lors du traitement de l'étape.

Pour afficher des fichiers journaux sur le nœud maître

1. Utilisez le protocole SSH pour vous connecter au nœud maître, comme décrit dans Connexion au nœud maître à l'aide de SSH (p. 368).
2. Accédez au répertoire qui contient les informations du fichier journal que vous souhaitez afficher. Le tableau précédent fournit une liste des types de fichiers journaux qui sont disponibles et leur emplacement. L'exemple suivant montre la commande permettant de naviguer dans le journal d'étape à l'aide d'un ID, `s-1234ABCDEFGH`.

```
cd /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/
```

3. Utilisez la visionneuse de fichier de votre choix pour afficher le fichier journal. L'exemple suivant utilise la commande `less` Linux pour afficher le fichier journal `controller`.

```
less controller
```

View Log Files Archived to Amazon S3

Par défaut, les clusters Amazon EMR lancés à l'aide de la console archivent automatiquement les fichiers journaux dans Amazon S3. Vous pouvez spécifier le chemin d'accès à votre propre journal ou autoriser la console à générer automatiquement un chemin d'accès au journal pour vous. Pour les clusters lancés à l'aide de l'interface de ligne de commande ou de l'API, vous devez configurer manuellement l'archivage des journaux Amazon S3.

Quand Amazon EMR est configuré pour archiver les fichiers journaux vers Amazon S3, il stocke les fichiers dans l'emplacement S3 que vous avez spécifié, dans le /*JobFlowId*/ dossier, où *JobFlowId* est l'identifiant du cluster.

Le tableau suivant répertorie quelques uns des fichiers journaux disponibles dans Amazon S3.

Emplacement	Description :
/*JobFlowId*node	Journaux de nœud, y compris les journaux d'action d'amorçage, d'état de l'instance et des applications pour le nœud. Les journaux de chaque nœud sont stockés dans un dossier identifié par l'identifiant de l'instance EC2 de ce nœud.
/*JobFlowId*node*instanceId*/*application*	Journaux créés par chaque application ou démon associé à une application. Par exemple, le journal du serveur Hive est situé dans *JobFlowId*/node/*instanceId*/hive/hive-server.log.
/*JobFlowId*steps*N*/	Journaux d'étape qui contiennent des informations sur le traitement de l'étape. La valeur de *N* indique le stéthoscope attribué par Amazon EMR. Par exemple, un cluster comporte deux étapes : s-1234ABCDEFGH et s-5678IJKLMNOP. La première étape est située dans /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ et la deuxième étape /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/. Les journaux d'étape suivants sont écrits par Amazon EMR. • contrôleur — Informations sur le traitement de l'étape. Si l'étape échoue lors de son chargement, vous pouvez trouver la trace de la pile dans ce journal. • syslog — Décrit l'exécution des tâches Hadoop dans l'étape.

Emplacement	Description :
	• stderr — Canal d'erreur standard de Hadoop lors du traitement de l'étape. • stdout — Canal de sortie standard de Hadoop lors du traitement de l'étape.
/*JobFlowId*/conteneurs	Journaux de conteneur d'applications. Les journaux pour chaque application YARN sont stockés à ces emplacements.
/*JobFlowId*/hadoop-mapreduce/	Journaux contenant des informations sur les détails de configuration et l'historique des tâches de MapReduce.

Pour afficher les fichiers journaux archivés dans Amazon S3 à l'aide de la console

1. Connectez-vous à AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Ouvrez le compartiment S3 spécifié lorsque vous avez configuré le cluster pour archiver les fichiers journaux dans Amazon S3.
3. Accédez au fichier journal qui contient les informations à afficher. Le tableau précédent fournit une liste des types de fichiers journaux qui sont disponibles et leur emplacement.
4. Double-cliquez sur un fichier journal pour l'afficher dans le navigateur.

Si vous ne voulez pas afficher les fichiers journaux dans la console Amazon S3, vous pouvez télécharger les fichiers de Amazon S3 vers votre ordinateur local à l'aide d'un outil tel que le plug-in Amazon S3 Organizer pour le navigateur web Firefox, ou en écrivant une application pour extraire les objets de Amazon S3. Pour plus d'informations, consultez Obtention d'objets dans le Amazon Simple Storage Service Manuel du développeur.

View Log Files in the Debugging Tool

Amazon EMR n'active pas automatiquement l'outil de débogage. Vous devez configurer celui-ci lorsque vous lancez le cluster.

Pour afficher les journaux du cluster à l'aide de la console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Dans la page Cluster List (Liste du cluster), choisissez l'icône détails en regard du cluster à afficher.

   La page Cluster Details (Détails du cluster) s'affiche. Dans la section Etapes, les liens situés à droite de chaque étape affichent les différents types de journaux disponibles pour l'étape. Ces journaux sont générés par Amazon EMR.
3. Pour afficher la liste des travaux Hadoop associés à une étape donnée, cliquez sur le lien View Jobs (Afficher les travaux) situé à droite de l'étape.
4. Pour afficher la liste des tâches Hadoop associées à un travail donné, cliquez sur le lien View Tasks (Afficher les tâches) situé à droite du travail.

5. Pour afficher la liste des tentatives exécutées par une tâche donnée lorsqu'elle essaie de prendre fin, cliquez sur le lien View Attemps (Afficher les tentatives) situé à droite de la tâche.

6. Pour afficher les journaux générés par une tentative de tâche, cliquez sur les liens stderr, stdout et syslog situés à droite de la tentative de tâche.

L'outil de débogage affiche les liens vers les fichiers journaux après le chargement par Amazon EMR des fichiers journaux dans votre compartiment sur Amazon S3. Etant donné que les fichiers journaux sont chargés sur Amazon S3 toutes les 5 minutes, les chargements des fichiers journaux peuvent prendre quelques minutes après la fin de l'étape.

Amazon EMR met à jour régulièrement l'état des travaux, des tâches et des tentatives de tâche Hadoop dans l'outil de débogage. Vous pouvez cliquer sur Refresh List (Actualiser la liste) dans les volets de débogage pour obtenir l'état le plus récent de ces éléments.

View Cluster Instances in Amazon EC2

Pour vous aider à gérer vos ressources, Amazon EC2 vous permet d'affecter des métadonnées à des ressources sous la forme de balises. Chaque balise Amazon EC2 se compose d'une clé et d'une valeur. Les balises vous permettent de classer vos ressources Amazon EC2 de différentes manières : par exemple, par objectif, par propriétaire ou par environnement.

Vous pouvez rechercher et filtrer des ressources en fonction des balises. Les balises assignées à l'aide de votre compte AWS sont disponibles uniquement pour vous. Les autres comptes partageant la ressource ne peuvent pas afficher vos balises.

Amazon EMR balise automatiquement chaque instance EC2 qu'il lance avec des paires clé-valeur qui identifient le cluster et le groupe d'instances auquel l'instance appartient. Il est ainsi facile de filtrer vos instances EC2 pour afficher, par exemple, uniquement les instances appartenant à un cluster spécifique ou pour afficher toutes les instances en cours d'exécution dans le groupe tâche-instance. Cela est particulièrement utile si vous exécutez plusieurs clusters simultanément ou que vous gérez un grand nombre d'instances EC2.

Il s'agit des paires clé-valeur prédéfinies qu'Amazon EMR attribue :

Clé	Value
aws:elasticmapreduce:job-flow-id	<job-flow-identifier>
aws:elasticmapreduce:instance-group-role	<group-role>

Les valeurs sont davantage définies comme suit :

- The <job-flow-identifier> is the ID of the cluster the instance is provisioned for. It appears in the format j-XXXXXXXXXXXX.
- The <group-role> is one of the following values: master, core, or task. These values correspond to the master instance group, core instance group, and task instance group.

Vous pouvez afficher et filtrer sur les balises qu'Amazon EMR ajoute. Pour plus d'informations, voir Utilisation des balises dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Parce que les balises définies par Amazon EMR sont des balises système et ne peuvent pas être modifiées ou supprimées, les sections d'affichage et de filtrage sont les plus pertinentes.

> **Note**
>
> Amazon EMR ajoute des balises à l'instance EC2 lorsque son état est mis à jour sur en cours d'exécution. S'il y a une période de latence entre le moment où l'instance EC2 est fournie et celui où son état est défini comme en cours d'exécution, les balises définies par Amazon EMR n'apparaissent pas jusqu'à ce que l'instance démarre. Si vous ne voyez pas les balises, attendez quelques minutes et actualisez la vue.

Événements et métriques CloudWatch

Vous pouvez utiliser des événements et des métriques pour suivre l'activité et l'état d'un cluster Amazon EMR, en affichant rapidement les événements et les métriques dans la console Amazon EMR pour un cluster unique, et en consultant des événements pour tous les clusters d'une région. Vous pouvez utiliser CloudWatch Events pour définir une action à effectuer quand Amazon EMR génère un événement qui correspond à un modèle que vous spécifiez, et également utiliser CloudWatch pour surveiller les métriques.

Les événements sont utiles pour surveiller une situation spécifique au sein d'un cluster, par exemple, quand un cluster passe de l'état de démarrage à celui d'exécution. Les métriques sont utiles pour surveiller une valeur spécifique, par exemple, le pourcentage d'espace disque disponible utilisé par HDFS au sein d'un cluster.

Pour de plus amples informations sur CloudWatch Events, veuillez consulter Guide de l'utilisateur Amazon CloudWatch Events. Pour plus d'informations sur les métriques CloudWatch, consultez Utilisation de métriques Amazon CloudWatch et Création d'alarmes Amazon CloudWatch dans le Guide de l'utilisateur Amazon CloudWatch.

Rubriques

- Supervision des événements CloudWatch (p. 339)
- Surveillance des métriques avec CloudWatch (p. 348)

Supervision des événements CloudWatch

Amazon EMR effectue le suivi des événements et conserve les informations les concernant sur les jusqu'à sept jours. Les changements d'état de clusters, de groupes d'instance, de stratégies de dimensionnement et d'étapes déclenchent l'enregistrement d'un événement. Chaque événement comporte des informations telles que la date et l'heure auxquelles l'événement s'est produit, ainsi que des détails supplémentaires sur l'événement, comme le cluster ou le groupe d'instances concerné.

Le tableau suivant répertorie des événements Amazon EMR, avec l'état ou le changement d'état indiqué par l'événement, la gravité de l'événement et les messages d'événement. Chaque événement est représenté sous la forme un objet JSON qui est automatiquement envoyé à un flux d'événements. L'objet JSON inclut des détails supplémentaires sur l'événement. L'objet JSON est particulièrement important lorsque vous configurez des règles pour le traitement des événements avec CloudWatch Events, car les règles visent à établir une correspondance avec des modèles dans l'objet JSON. Pour plus d'informations, consultez Événements et modèles d'événements et Événements Amazon EMR dans le Guide de l'utilisateur Amazon CloudWatch Events.

Événements de cluster

État ou changement d'état	Gravité	Message
`STARTING`	`INFO`	cluster Amazon EMR *`ClusterId (ClusterName)`* a été demandé à *`Time`* et est en cours de création.
`STARTING`	`INFO`	**Note** S"applique uniquement aux clusters avec la configuration des parcs d'instances et plusieurs sous-réseaux sélectionnés au sein d'un VPC. cluster Amazon EMR *`ClusterId (ClusterName)`* est en cours de création dans le sous-réseau (*`SubnetName`*) dans le VPC (*`VPCName`*) dans la zone de disponibilité (*`AvailabilityZoneID`*), qui a été choisi parmi les options VPC spécifiées.
`STARTING`	`INFO`	**Note** S'applique uniquement aux clusters avec la configuration des parcs d'instances et plusieurs zones de disponibilité

État ou changement d'état	Gravité	Message
		sélectionnées au sein de EC2-Classic. cluster Amazon EMR *ClusterId (ClusterName)* est en cours de création dans la zone de disponibilité (*AvailabilityZoneID*), qui a été choisi parmi les options de zone de disponibilité spécifiées.
`RUNNING`	`INFO`	cluster Amazon EMR *ClusterId (ClusterName)* a commencé à exécuter des étapes à *Time*.
`WAITING`	`INFO`	cluster Amazon EMR *ClusterId (ClusterName)* a été créé à *Time* et est prêt à l'emploi. —ou— cluster Amazon EMR *ClusterId (ClusterName)* terminé l'exécution de toutes les étapes en attente à *Time*. Note Un cluster à l'état `WAITING` peut néanmoins traiter des tâches.
`TERMINATED`	La gravité dépend de la raison du changement d'état, comme illustré dans les exemples suivants : • **`CRITICAL`** si le cluster s'est terminé avec l'un des motifs de changement d'état suivants : `INTERNAL_ERROR`, `VALIDATION_ERROR`, `INSTANCE_FAILURE`, `BOOTSTRAP_FAILURE`ou `STEP_FAILURE`. • **`INFO`** si le cluster s'est terminé avec l'un des motifs de changement d'état suivants : `USER_REQUEST` ou `ALL_STEPS_COMPLETED`.	Cluster Amazon EMR *ClusterId (ClusterName)* a pris fin à *Time* avec un motif de *StateChangeReason:Code*.

État ou changement d'état	Gravité	Message
`TERMINATED_WITH_ERRORS`	`CRITICAL`	Cluster Amazon EMR *`ClusterId (ClusterName)`* s'est terminé avec des erreurs à *`Time`* avec un motif de *`StateChangeReason:Code`*.

Événements de parc d'instances

> **Note**
>
> La configuration des parcs d'instances est disponible uniquement dans les versions 4.8.0 et ultérieures d'Amazon EMR, à l'exception des versions 5.0.0 et 5.0.3.

État ou changement d'état	Gravité	Message
À partir de `PROVISIONING` à `WAITING`	`INFO`	Provisionnement pour le parc d'instances *`InstanceFleetID`* dans Amazon EMR du cluster *`ClusterId (ClusterName)`* est terminé. Approvisionnement démarré à *`Time`* et pris *`Num`* minutes. Le parc d'instances dispose désormais d'une capacité à la demande de *`Num`* et capacité Spot de *`Num`*. La capacité cible à la demande était de *`Num`*et la capacité Spot cible était de *`Num`*.
À partir de `WAITING` à `RESIZING`	`INFO`	Un redimensionnement pour le parc d'instances *`InstanceFleetID`* dans le cluster Amazon EMR *`ClusterId (ClusterName)`* a commencé à *`Time`*. Le parc d'instances est redimensionné à partir d'une capacité à la demande de *`Num`* à une cible de *`Num`*, et à partir d'une capacité Spot de *`Num`* à une cible de *`Num`*.
À partir de `RESIZING` à `WAITING`	`INFO`	L'opération de redimensionnement pour le parc d'instances *`InstanceFleetID`* dans le cluster Amazon EMR *`ClusterId (ClusterName)`* est terminé. Le redimensionnement a commencé à *`Time`* et pris *`Num`* minutes. Le parc d'instances dispose désormais d'une capacité à la demande de *`Num`* et capacité Spot de *`Num`*. La capacité cible

État ou changement d'état	Gravité	Message
		à la demande était de *Num* et la capacité Spot cible était *Num*.
À partir de RESIZING à WAITING	WARN	L'opération de redimensionnement pour le parc d'instances *InstanceFleetID* dans le cluster Amazon EMR *ClusterId (ClusterName)* a atteint le délai d'expiration et a arrêté. Le redimensionnement a commencé à *Time* et arrêté après *Num* minutes. Le parc d'instances dispose désormais d'une capacité à la demande de *Num* et capacité Spot de *Num*. La capacité cible à la demande était de *Num* et la capacité Spot cible était *Num*.
SUSPENDED	ERROR	Parc d'instances *InstanceFleetID* dans Amazon EMR du cluster *ClusterId (ClusterName)* a été arrêté à *Time* pour la raison suivante : *ReasonDesc*.
RESIZING	WARNING	L'opération de redimensionnement pour le parc d'instances *InstanceFleetID* dans Amazon EMR du cluster *ClusterId (ClusterName)* est bloqué pour la raison suivante : *ReasonDesc*.
WAITING ou RUNNING	INFO	Un redimensionnement pour le parc d'instances *InstanceFleetID* dans Amazon EMR du cluster *ClusterId (ClusterName)* a été initié par *Entity* à *Time*.

Événements de groupe d'instances

État ou changement d'état	Gravité	Message
À partir de RESIZING à RUNNING	INFO	L'opération de redimensionnement pour le groupe d'instances *InstanceGroupID* dans Amazon EMR du cluster *ClusterId (ClusterName)* est terminé. Il a maintenant un nombre d'instances de *Num*. Le redimensionnement a commencé

État ou changement d'état	Gravité	Message
		à *Time* et pris *Num* minutes pour terminer.
À partir de `RUNNING` à `RESIZING`	`INFO`	Un redimensionnement pour le groupe d'instances *InstanceGroupID* dans Amazon EMR du cluster *ClusterId (ClusterName)* a commencé à *Time*. Il est redimensionné à partir d'un nombre d'instances de *Num* to *Num*.
`SUSPENDED`	`ERROR`	Groupe d'instances *InstanceGroupID* dans Amazon EMR du cluster *ClusterId (ClusterName)* a été arrêté à *Time* pour la raison suivante : *ReasonDesc*.
`RESIZING`	`WARNING`	L'opération de redimensionnement pour le groupe d'instances *InstanceGroupID* dans Amazon EMR du cluster *ClusterId (ClusterName)* est bloqué pour la raison suivante : *ReasonDesc*.
À partir de `RUNNING` à `RESIZING`	`INFO`	Un redimensionnement pour le groupe d'instances *InstanceGroupID* dans Amazon EMR du cluster *ClusterId (ClusterName)* a été initié par *Entity* à *Time*.

> **Note**
>
> Avec Amazon EMR version 5.21.0 ou ultérieure, vous pouvez remplacer les configurations de cluster et spécifier des classifications de configuration supplémentaires pour chaque groupe d'instances d'un cluster en cours d'exécution. Vous pouvez effectuer cette opération à l'aide de la console Amazon EMR, de l'AWS Command Line Interface (AWS CLI) ou du kit SDK AWS. Pour de plus amples informations, veuillez consulter Fourniture d'une configuration pour un groupe d'instances dans un cluster en cours d'exécution.

Le tableau suivant répertorie des événements Amazon EMR liés à l'opération de reconfiguration, avec l'état ou le changement d'état indiqué par l'événement, la gravité de l'événement et les messages d'événement.

État ou changement d'état	Gravité	Message
`RUNNING`	`INFO`	Une reconfiguration pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* a

État ou changement d'état	Gravité	Message
		été initié par l'utilisateur à *Time*. La version de la configuration demandée est *Num*.
À partir de RECONFIGURING à RUNNING	INFO	L'opération de reconfiguration pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* est terminé. La reconfiguration a commencé à *Time* et pris *Num* minutes pour terminer. La version de configuration actuelle est *Num*.
À partir de RUNNING à RECONFIGURING	INFO	Une reconfiguration pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* a commencé à *Time*. Il est en cours de configuration à partir du numéro de version *Num* vers le numéro de version *Num*.
RESIZING	INFO	Opération de reconfiguration vers la version de configuration *Num* pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* est temporairement bloqué à *Time* car le groupe d'instances est dans *State*.
RECONFIGURING	INFO	Opération de redimensionnement du nombre d'instances *Num* pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* est temporairement bloqué à *Time* car le groupe d'instances est dans *State*.
RECONFIGURING	WARNING	L'opération de reconfiguration pour le groupe d'instances *InstanceGroupID* dans le Amazon EMR du cluster *ClusterId (ClusterName)* a échoué à *Time* et pris *Num* minutes avant l'échec. La version de configuration a échoué *Num*.

État ou changement d'état	Gravité	Message
`RECONFIGURING`	`INFO`	Les configurations reprennent leur numéro de version précédent *`Num`*pour le groupe d'instances *`InstanceGroupID`* dans le Amazon EMR du cluster *`ClusterId (ClusterName)`* à *`Time`*. La nouvelle version de configuration est *`Num`*.
À partir de `RECONFIGURING` à `RUNNING`	`INFO`	Les configurations ont été rétablies avec succès à la version précédente *`Num`* pour le groupe d'instances *`InstanceGroupID`* dans le Amazon EMR du cluster *`ClusterId (ClusterName)`* à *`Time`*. La nouvelle version de configuration est *`Num`*.
À partir de `RECONFIGURING` à `SUSPENDED`	`CRITICAL`	Impossible de revenir à la version précédente réussie *`Num`* pour Groupe d'instances *`InstanceGroupID`* dans le Amazon EMR du cluster *`ClusterId (ClusterName)`* à *`Time`*.

Événements de stratégie de dimensionnement automatique

État ou changement d'état	Gravité	Message
`PENDING`	`INFO`	Une stratégie Auto Scaling a été ajoutée au groupe d'instances *`InstanceGroupID`* dans Amazon EMR du cluster *`ClusterId (ClusterName)`* à *`Time`*. La stratégie est en attente de la pièce jointe. —ou— Stratégie Auto Scaling pour le groupe d'instances *`InstanceGroupID`* dans Amazon EMR du cluster *`ClusterId (ClusterName)`* a été mis à jour à *`Time`*. La stratégie est en attente de la pièce jointe.
`ATTACHED`	`INFO`	Stratégie Auto Scaling pour le groupe d'instances *`InstanceGroupID`* dans Amazon EMR du cluster

État ou changement d'état	Gravité	Message
		`ClusterId (ClusterName)` a été joint à `Time`.
`DETACHED`	`INFO`	Stratégie Auto Scaling pour le groupe d'instances `InstanceGroupID` dans Amazon EMR du cluster `ClusterId (ClusterName)` a été détaché à `Time`.
`FAILED`	`ERROR`	Stratégie Auto Scaling pour le groupe d'instances `InstanceGroupID` dans Amazon EMR du cluster `ClusterId (ClusterName)` n'a pas pu être attaché et a échoué à `Time`. —ou— Stratégie Auto Scaling pour le groupe d'instances `InstanceGroupID` dans Amazon EMR du cluster `ClusterId (ClusterName)` n'a pas pu se détacher et a échoué à `Time`.

Événements d'étape

État ou changement d'état	Gravité	Message
`PENDING`	`INFO`	Étape `StepID (StepName)` a été ajouté à Amazon EMR du cluster `ClusterId (ClusterName)` à `Time` et est en attente d'exécution.
`CANCEL_PENDING`	`WARN`	Étape `StepID (StepName)` dans Amazon EMR du cluster `ClusterId (ClusterName)` a été annulé à `Time` et est en attente d'annulation.
`RUNNING`	`INFO`	Étape `StepID (StepName)` dans Amazon EMR du cluster `ClusterId (ClusterName)` a commencé à s'exécuter à `Time`.
`COMPLETED`	`INFO`	Étape `StepID (StepName)` dans Amazon EMR du cluster `ClusterId (ClusterName)` a terminé l'exécution à `Time`. L'étape a commencé à s'exécuter

État ou changement d'état	Gravité	Message
		à *Time* et pris *Num* minutes pour terminer.
CANCELLED	WARN	La demande d'annulation a réussi pour l'étape du cluster *StepID (StepName)* dans Amazon EMR du cluster *ClusterId (ClusterName)* à *Time*, et l'étape est maintenant annulée.
FAILED	ERROR	Étape *StepID (StepName)* dans Amazon EMR du cluster *ClusterId (ClusterName)* a échoué à *Time*.

Affichage des événements à l'aide de la console Amazon EMR

Pour chaque cluster, vous pouvez consulter une liste simple d'événements dans le volet des détails, qui répertorie les événements par ordre décroissant d'occurrence. Vous pouvez également afficher tous les événements pour tous les clusters d'une région par ordre décroissant d'occurrence.

> **Note**
>
> Si vous ne souhaitez pas qu'un utilisateur voit tous les événements de cluster pour une région, ajoutez une instruction qui refuse l'autorisation (`"Effect": "Deny"`) pour l'action `elasticmapreduce:ViewEventsFromAllClustersInConsole` à une stratégie attachée à l'utilisateur.

Pour afficher tous les événements pour tous les clusters d'une région

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Events (Événements).

Pour afficher les événements pour un cluster particulier

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Liste de clusters, sélectionnez un cluster, puis choisissez Afficher les détails.
3. Choisissez Events (Événements) dans le volet des détails du cluster.

Création de règles pour des événements Amazon EMR avec CloudWatch

Amazon EMR envoie automatiquement les événements à un flux d'événements CloudWatch. Vous pouvez créer des règles qui correspondent à des événements selon un modèle spécifié, et acheminer les événements vers des cibles pour prendre des mesures, comme envoyer une notification par e-mail. Les modèles sont mis en correspondance avec l'objet JSON d'événement. Pour de plus amples informations sur les détails d'événement Amazon EMR, veuillez consulter Événements Amazon EMR dans le Guide de l'utilisateur Amazon CloudWatch Events.

Pour en savoir plus sur la configuration de règles d'événement CloudWatch, consultez la page Création d'une règle CloudWatch qui se déclenche sur un événement.

Surveillance des métriques avec CloudWatch

Les métriques sont mises à jour toutes les cinq minutes, et automatiquement collectées et transmises à CloudWatch pour chaque cluster EMR. Cet intervalle n'est pas configurable. Les métriques Amazon EMR rapportées dans CloudWatch ne sont pas facturées. Les métriques sont archivées pendant deux semaines, après quoi les données sont supprimées.

Comment utiliser les métriques Amazon EMR ?

Les métriques présentées par Amazon EMR fournissent des informations qui permettent divers types d'analyses. Le tableau ci-dessous présente certaines utilisations courantes des métriques. Voici quelques suggestions pour vous aider à démarrer, qui ne constituent pas une liste exhaustive. Pour consulter une liste complète des métriques présentées par Amazon EMR, consultez Métriques présentées par Amazon EMR dans CloudWatch (p. 352).

Comment... ?	Métriques pertinentes
Suivre la progression de mon cluster	Regardez le `RunningMapTasks`, `RemainingMapTasks`, `RunningReduceTasks`, et `RemainingReduceTasks` les métriques.
Détecter les clusters inactifs	La métrique `IsIdle` vérifie si un cluster est présent mais n'exécute actuellement aucune tâche. Vous pouvez définir une alarme afin qu'elle se

Comment... ?	Métriques pertinentes
	déclenche lorsque le cluster est inactif pendant une période donnée, par exemple 30 minutes.
Détecter lorsqu'un nœud ne dispose plus d'un espace de stockage suffisant	La métrique `HDFSUtilization` représente le pourcentage d'espace disque actuellement utilisé. Si ce chiffre dépasse un niveau acceptable pour votre application, par exemple 80 % de capacité utilisée, vous devrez peut-être redimensionner votre cluster et ajouter des nœuds principaux.

Accès aux métriques CloudWatch

Il existe de nombreuses façons d'accéder aux métriques qu'Amazon EMR transmet à CloudWatch. Vous pouvez les afficher via la console Amazon EMR ou la console CloudWatch, ou les récupérer à l'aide de l'interface de ligne de commande CloudWatch ou de l'API CloudWatch. Les procédures suivantes vous montrent comment accéder aux métriques à l'aide de ces différentes outils.

Pour afficher les métriques dans la console Amazon EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Pour afficher les métriques pour un cluster, sélectionnez un cluster pour afficher le volet Récapitulatif.
3. Choisissez Surveillance afin d'afficher les informations sur ce cluster. Choisissez l'un des onglets Statut du cluster, Mapper/Réduire, Statut du nœud, E/S ou HBase pour charger les rapports sur la progression et l'état du cluster.
4. Une fois que vous avez choisi une métrique à afficher, vous pouvez sélectionner une taille de graphique. Modifiez les champs Start (Démarrer) et End (Terminer) pour filtrer les métriques sur une période de temps spécifique.

Pour afficher les métriques dans la console CloudWatch

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez EMR.
3. Faites défiler jusqu'à la métrique que vous souhaitez représenter graphiquement. Vous pouvez rechercher par l'identifiant de cluster du cluster à surveiller.

4. Ouvrez une métrique pour afficher le graphique.

Pour accéder aux métriques à partir de l'interface de ligne de commande CloudWatch

- Appeler `mon-get-stats`. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon CloudWatch.

Pour accéder aux métriques à partir de l'API CloudWatch

- Appeler `GetMetricStatistics`. Pour plus d'informations, consultez Amazon CloudWatch API Reference.

Configuration d'alarmes sur les métriques

Amazon EMR envoie les métriques à CloudWatch, ce qui signifie que vous pouvez utiliser CloudWatch pour configurer des alarmes sur vos métriques Amazon EMR. Vous pouvez, par exemple, configurer une alarme dans CloudWatch pour qu'un e-mail vous soit envoyé dès que l'utilisation HDFS est supérieure à 80 %.

Les rubriques suivantes vous offrent une présentation générale de la configuration des alarmes à l'aide de CloudWatch. Pour obtenir des instructions détaillées, consultez Création ou modification d'une alarme CloudWatch dans le Guide de l'utilisateur Amazon CloudWatch.

Définir des alarmes à l'aide de la console CloudWatch

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Sélectionnez Create Alarm. Cela fait démarrer l'Create Alarm Wizard (Assistant de création d'alarme).
3. Choisissez Métriques EMR et faites défiler les métriques Amazon EMR afin de localiser la métrique sur laquelle vous voulez placer une alarme. Pour afficher uniquement les métriques Amazon EMR dans cette boîte de dialogue, il suffit d'effectuer une recherche sur l'identifiant de cluster de votre cluster. Sélectionnez la métrique sur laquelle créer une alarme, puis sélectionnez Suivant.
4. Remplissez les valeurs des champs Name (Nom), Description, Threshold (Seuil) et Time (Heure) pour la métrique.
5. Si vous voulez que CloudWatch vous envoie un e-mail lorsque l'état d'alarme est atteint, choisissez État d'ALARME dans le champ Lorsque cette alarme :. Pour champ Send notification to (Envoyer une notification à), sélectionnez une rubrique SNS existante. Si vous choisissez Create topic (Créer une rubrique), vous pouvez définir le nom d'une nouvelle liste d'abonnement par e-mail et les adresses e-mail pour cette liste. La liste est enregistrée et s'affiche dans le champ des alarmes futures.

   Note

   Si vous utilisez Créer la rubrique pour créer une nouvelle rubrique Amazon SNS, les adresses e-mail doivent être vérifiées avant de pouvoir recevoir des notifications. Les e-mails sont envoyés uniquement lorsque l'alarme passe à un état défini. Si ce changement d'état de l'alarme se produit avant la vérification des adresses e-mail, ces dernières ne reçoivent pas de notification.
6. A ce stade, l'écran Define Alarm (Définir une alarme) vous offre la possibilité de vérifier l'alarme que vous être sur le point de créer. Sélectionnez Create Alarm.

   Note

   Pour plus d'informations sur la configuration des alarmes à l'aide de la console CloudWatch, consultez Créer une alarme qui envoie un e-mail dans le Guide de l'utilisateur Amazon CloudWatch.

Pour définir une alarme à l'aide de l'API CloudWatch

- Appeler `mon-put-metric-alarm`. Pour plus d'informations, consultez Guide de l'utilisateur Amazon CloudWatch.

Pour définir une alarme à l'aide de l'API CloudWatch

- Appeler `PutMetricAlarm`. Pour plus d'informations, consultez Amazon CloudWatch API Reference

Métriques présentées par Amazon EMR dans CloudWatch

Les tableaux suivant répertorient toutes les métriques présentées par Amazon EMR dans la console et transmises à CloudWatch.

Métriques de Amazon EMR

Amazon EMR envoie les données de plusieurs métriques vers CloudWatch. Tous les clusters Amazon EMR envoient automatiquement des métriques à intervalles de cinq minutes. Les métriques sont archivées pendant deux semaines ; après cette période, les données sont supprimées.

L'espace de noms `AWS/ElasticMapReduce` inclut les métriques suivantes.

> **Note**
>
> Amazon EMR extrait les métriques d'un cluster. Si un cluster devient inaccessible, aucune des métriques n'est signalée jusqu'à ce que le cluster redevienne disponible.

Les métriques suivantes sont disponibles pour les clusters exécutant les versions 2.x de Hadoop.

Métrique	Description
Statut du cluster	
IsIdle	Indique qu'un cluster ne s'exécute plus, mais est encore en actif et génère des frais. Il est défini sur 1 si aucune tâche ni aucun travail n'est en cours d'exécution, et défini sur 0 dans le cas contraire. Cette valeur est vérifiée à intervalles de cinq minutes et une valeur de 1 indique uniquement que le cluster a été inactif lors de la vérification, et non pas qu'il a été inactif pendant les cinq minutes entières. Pour éviter les fausses erreurs, vous devez déclencher une alarme lorsque cette valeur est 1 pendant plusieurs contrôles consécutifs de 5 minutes. Par exemple, vous pouvez déclencher une alarme pour cette valeur si elle renvoie 1 pendant au moins 30 minutes. Cas d'utilisation : Moniteur cluster performance Unités: Booléen
ContainerAllocated	Nombre de conteneurs de ressources attribués par ResourceManager. Cas d'utilisation : Moniteur cluster progression Unités: Count
ContainerReserved	Nombre de conteneurs réservés. Cas d'utilisation : Moniteur cluster progression Unités: Count
ContainerPending	Nombre de conteneurs dans la file d'attente qui n'ont pas encore été alloués. Cas d'utilisation : Moniteur cluster progression Unités: Count
ContainerPendingRatio	Le rapport entre les conteneurs en attente et les conteneurs attribués (Rapport de conteneurs en attente = Conteneurs en attente/Conteneurs attribués). Si les Conteneurs attribués = 0, le Rapport des conteneurs en attente = Conteneurs en attente. La valeur du Rapport des conteneurs en attente est exprimée sous forme de nombre et non de pourcentage. Cette valeur est utile pour dimensionner les ressources de cluster en fonction du comportement d'attribution des conteneurs. Unités: Count
AppsCompleted	Nombre de demandes soumises à YARN ayant été traitées.

Métrique	Description
	Cas d’utilisation : Moniteur cluster progression Unités: Count
AppsFailed	Nombre de demandes soumises à YARN impossibles à traiter. Cas d’utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
AppsKilled	Nombre d'applications soumises à YARN ayant été désactivées. Cas d’utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
AppsPending	Nombre d'applications soumises à YARN qui se trouvent dans un état d'attente. Cas d’utilisation : Moniteur cluster progression Unités: Count
AppsRunning	Nombre d'applications soumises à YARN qui sont en cours d'exécution. Cas d’utilisation : Moniteur cluster progression Unités: Count
AppsSubmitted	Nombre d'applications soumises à YARN. Cas d’utilisation : Moniteur cluster progression Unités: Count
Statut du nœud	
CoreNodesRunning	Nombre de nœuds principaux actifs. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d’utilisation : Moniteur cluster santé Unités: Count
CoreNodesPending	Nombre de nœuds principaux en attente d'attribution. Il se peut que tous les nœuds principaux demandés ne soient pas immédiatement accessibles ; cette métrique indique les demandes en attente. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d’utilisation : Moniteur cluster santé Unités: Count

Métrique	Description
LiveDataNodes	Pourcentage de nœuds de données qui reçoivent des tâches de Hadoop. Cas d'utilisation : Moniteur cluster santé Unités: Pourcentage
MRTotalNodes	Nombre de nœuds actuellement disponibles pour les tâches MapReduce. Équivalent à la métrique YARN `mapred.resourcemanager.TotalNodes`. Utiliser comme : Moniteur cluster progression Unités: Count
MRActiveNodes	Nombre de nœuds exécutant actuellement des tâches ou travaux MapReduce. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfActiveNodes`. Cas d'utilisation : Moniteur cluster progression Unités: Count
MRLostNodes	Nombre de nœuds alloués à MapReduce qui ont été marqués à l'état PERDU. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfLostNodes`. Cas d'utilisation : Moniteur cluster état de santé, Moniteur cluster progression Unités: Count
MRUnhealthyNodes	Nombre de nœuds disponibles pour les tâches MapReduce marqués à l'état NON SAIN. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfUnhealthyNodes`. Cas d'utilisation : Moniteur cluster progression Unités: Count
MRDecommissionedNodes	Nombre de nœuds alloués aux applications MapReduce qui ont été marqués à l'état HORS SERVICE. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfDecommissionedNodes`. Utiliser comme : Moniteur cluster état de santé, Moniteur cluster progression Unités: Count

Métrique	Description
MRRebootedNodes	Nombre de nœuds disponibles pour MapReduce ayant été redémarrés et marqués à l'état REINITIALISE. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfRebootedNodes`. Cas d'utilisation : Moniteur cluster état de santé, Moniteur cluster progression Unités: Count
MultiMasterInstanceGroupNodesRunning	Le nombre de nœuds maîtres en cours d'exécution. Cas d'utilisation : Surveillance de la défaillance et du remplacement du nœud maître Unités: Count
MultiMasterInstanceGroupNodesRunningPercentage	Le pourcentage de nœuds maîtres en cours d'exécution sur le nombre d'instances de nœuds maîtres demandées. Cas d'utilisation : Surveillance de la défaillance et du remplacement du nœud maître Unités: Pourcentage
MultiMasterInstanceGroupNodesRequested	Le nombre de nœuds maîtres demandés. Cas d'utilisation : Surveillance de la défaillance et du remplacement du nœud maître Unités: Count
E/S	
S3BytesWritten	Nombre d'octets écrits sur Amazon S3. Cas d'utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
S3BytesRead	Nombre d'octets lus à partir de Amazon S3. Cas d'utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
HDFSUtilization	Pourcentage de stockage HDFS actuellement utilisé. Cas d'utilisation : Analyser cluster performance Unités: Pourcentage

Métrique	Description
HDFSBytesRead	Nombre d'octets lus à partir de HDFS. Cette métrique regroupe uniquement les tâches MapReduce et ne s'applique pas aux autres charges de travail sur EMR. Cas d'utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
HDFSBytesWritten	Nombre d'octets écrits sur HDFS. Cette métrique regroupe uniquement les tâches MapReduce et ne s'applique pas aux autres charges de travail sur EMR. Cas d'utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
MissingBlocks	Nombre de blocs dans lesquels HDFS n'a aucun réplica. Il peut s'agir de blocs corrompus. Cas d'utilisation : Moniteur cluster santé Unités: Count
CorruptBlocks	Nombre de blocs que HDFS indique comme étant corrompus. Cas d'utilisation : Moniteur cluster santé Unités: Count
TotalLoad	Nombre total de transferts de données simultanés. Cas d'utilisation : Moniteur cluster santé Unités: Count
MemoryTotalMB	Quantité totale de mémoire dans le cluster. Cas d'utilisation : Moniteur cluster progression Unités: Count
MemoryReservedMB	Quantité de mémoire réservée. Cas d'utilisation : Moniteur cluster progression Unités: Count
MemoryAvailableMB	Quantité de mémoire disponible à allouer. Cas d'utilisation : Moniteur cluster progression Unités: Count

Métrique	Description
YARNMemoryAvailablePercentage	Le pourcentage de mémoire disponible restante pour YARN (Pourcentage de mémoire disponible YARN = Mémoire disponible en Mo/Total de mémoire en Mo). Cette valeur est utile pour dimensionner les ressources de cluster en fonction de l'utilisation de mémoire YARN. Unités: Pourcentage
MemoryAllocatedMB	Quantité de mémoire allouée au cluster. Cas d'utilisation : Moniteur cluster progression Unités: Count
PendingDeletionBlocks	Nombre de blocs marqués pour la suppression. Cas d'utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
UnderReplicatedBlocks	Nombre de blocs devant être répliqués une ou plusieurs fois. Cas d'utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
DfsPendingReplicationBlocks	État de la réplication des blocs : blocs en cours de réplication, l'âge des demandes de réplication et demandes de réplication ayant échoué. Cas d'utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
CapacityRemainingGB	Quantité de capacité du disque HDFS restante. Cas d'utilisation : Moniteur cluster progression, Surveillance cluster santé Unités: Count
HBase	
HbaseBackupFailed	Si la dernière sauvegarde a échoué. La valeur est définie sur 0 par défaut et mise à jour sur 1 en cas d'échec de la tentative de sauvegarde précédente. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveiller les sauvegardes HBase Unités: Count

Métrique	Description
MostRecentBackupDuration	Délai nécessaire à l'exécution de la précédente sauvegarde. Cette métrique est définie même si la dernière sauvegarde a réussi ou a échoué. Lorsque la sauvegarde est en cours, cette métrique retourne le nombre de minutes qui se sont écoulées depuis le démarrage de la sauvegarde. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveillance des sauvegardes HBase Unités: minutes
TimeSinceLastSuccessfulBackup	Nombre de minutes écoulées depuis la dernière sauvegarde de HBase réussie démarrée sur votre cluster. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveiller les sauvegardes HBase Unités: minutes

Voici les métriques Hadoop 1 :

Métrique	Description
Statut du cluster	
IsIdle	Indique qu'un cluster ne s'exécute plus, mais est encore en actif et génère des frais. Il est défini sur 1 si aucune tâche ni aucun travail n'est en cours d'exécution, et défini sur 0 dans le cas contraire. Cette valeur est vérifiée à intervalles de cinq minutes et une valeur de 1 indique uniquement que le cluster a été inactif lors de la vérification, et non pas qu'il a été inactif pendant les cinq minutes entières. Pour éviter les fausses erreurs, vous devez déclencher une alarme lorsque cette valeur est 1 pendant plusieurs contrôles consécutifs de 5 minutes. Par exemple, vous pouvez déclencher une alarme pour cette valeur si elle renvoie 1 pendant au moins 30 minutes. Cas d'utilisation : Moniteur cluster performance Unités: Booléen
JobsRunning	Nombre de tâches en cours d'exécution dans le cluster. Cas d'utilisation : Moniteur cluster santé Unités: Count
JobsFailed	Nombre de tâches qui ont échoué dans le cluster. Cas d'utilisation : Moniteur cluster santé Unités: Count
Mapper/Réduire	

Métrique	Description
MapTasksRunning	Nombre de tâches de mappage en cours d'exécution pour chaque tâche. Si un planificateur est installé et plusieurs tâches sont en cours d'exécution, plusieurs graphiques sont générés. Cas d'utilisation : Moniteur cluster progression Unités: Count
MapTasksRemaining	Nombre de tâches de mappage restantes pour chaque tâche. Si un planificateur est installé et plusieurs tâches sont en cours d'exécution, plusieurs graphiques sont générés. Une tâche de mappage restante est une tâche qui n'est pas dans l'un des états suivants : En cours d'exécution, Tué ou Terminé. Cas d'utilisation : Moniteur cluster progression Unités: Count
MapSlotsOpen	Capacité de tâche de mappage inutilisée. Elle est calculée sur la base du nombre maximal de tâches de mappage pour un cluster donné, moins le nombre total de tâches de mappage en cours d'exécution dans ce cluster. Cas d'utilisation : Analyser cluster performance Unités: Count
RemainingMapTasksPerSlot	Rapport entre les tâches de mappage total restantes et le nombre total d'emplacements de mappage disponibles dans le cluster. Cas d'utilisation : Analyser cluster performance Unités: Ratio
ReduceTasksRunning	Nombre de tâches de réduction en cours d'exécution pour chaque tâche. Si un planificateur est installé et plusieurs tâches sont en cours d'exécution, plusieurs graphiques sont générés. Cas d'utilisation : Moniteur cluster progression Unités: Count
ReduceTasksRemaining	Nombre de tâches de réduction restantes pour chaque tâche. Si un planificateur est installé et plusieurs tâches sont en cours d'exécution, plusieurs graphiques sont générés. Cas d'utilisation : Moniteur cluster progression Unités: Count

Métrique	Description
ReduceSlotsOpen	Capacité des tâches de réduction inutilisée. Elle est calculée sur la base de la capacité des tâches de réduction maximale pour un cluster donné, moins le nombre de tâches de réduction en cours d'exécution dans ce cluster. Cas d'utilisation : Analyser cluster performance Unités: Count
Statut du nœud	
CoreNodesRunning	Nombre de nœuds principaux actifs. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d'utilisation : Moniteur cluster santé Unités: Count
CoreNodesPending	Nombre de nœuds principaux en attente d'attribution. Il se peut que tous les nœuds principaux demandés ne soient pas immédiatement accessibles ; cette métrique indique les demandes en attente. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d'utilisation : Moniteur cluster santé Unités: Count
LiveDataNodes	Pourcentage de nœuds de données qui reçoivent des tâches de Hadoop. Cas d'utilisation : Moniteur cluster santé Unités: Pourcentage
TaskNodesRunning	Nombre de nœuds de tâches actifs. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d'utilisation : Moniteur cluster santé Unités: Count
TaskNodesPending	Nombre de nœuds de tâches en attente d'attribution. Il se peut que tous les nœuds de tâches demandés ne soient pas immédiatement accessibles ; cette métrique indique les demandes en attente. Les points de données pour cette métrique sont présentés uniquement s'il existe un groupe d'instances correspondant. Cas d'utilisation : Moniteur cluster santé Unités: Count

Métrique	Description
LiveTaskTrackers	Pourcentage de dispositifs de suivi des tâches fonctionnels. Cas d’utilisation : Moniteur cluster santé Unités: Pourcentage
E/S	
S3BytesWritten	Nombre d'octets écrits sur Amazon S3. Cette métrique regroupe uniquement les tâches MapReduce et ne s'applique pas aux autres charges de travail sur EMR. Cas d’utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
S3BytesRead	Nombre d'octets lus à partir de Amazon S3. Cette métrique regroupe uniquement les tâches MapReduce et ne s'applique pas aux autres charges de travail sur EMR. Cas d’utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
HDFSUtilization	Pourcentage de stockage HDFS actuellement utilisé. Cas d’utilisation : Analyser cluster performance Unités: Pourcentage
HDFSBytesRead	Nombre d'octets lus à partir de HDFS. Cas d’utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
HDFSBytesWritten	Nombre d'octets écrits sur HDFS. Cas d’utilisation : Analyser cluster performance, surveillance cluster progression Unités: Count
MissingBlocks	Nombre de blocs dans lesquels HDFS n'a aucun réplica. Il peut s'agir de blocs corrompus. Cas d’utilisation : Moniteur cluster santé Unités: Count

Métrique	Description
TotalLoad	Nombre total actuel de lecteurs et auteurs indiqué par tous les DataNodes d'un cluster. Cas d'utilisation : de diagnostiquer le degré auquel des E/S élevées peuvent contribuer à de mauvaises performances d'exécution des tâches ; Les nœuds de travail exécutant le démon DataNode doivent également procéder à des tâches de mappage et de réduction. Des valeurs TotalLoad constamment élevées peuvent être le signe que des I/O élevées contribuent à des performances médiocres. Des pics occasionnels de cette valeur ne sont pas inhabituels et ne sont généralement pas le signe d'un problème. Unités: Count
HBase	
BackupFailed	Si la dernière sauvegarde a échoué. La valeur est définie sur 0 par défaut et mise à jour sur 1 en cas d'échec de la tentative de sauvegarde précédente. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveiller les sauvegardes HBase Unités: Count
MostRecentBackupDuration	Délai nécessaire à l'exécution de la précédente sauvegarde. Cette métrique est définie même si la dernière sauvegarde a réussi ou a échoué. Lorsque la sauvegarde est en cours, cette métrique retourne le nombre de minutes qui se sont écoulées depuis le démarrage de la sauvegarde. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveillance des sauvegardes HBase Unités: minutes
TimeSinceLastSuccessfulBackup	Nombre de minutes écoulées depuis la dernière sauvegarde de HBase réussie démarrée sur votre cluster. Cette mesure est présentée uniquement pour les clusters. HBase. Cas d'utilisation : Surveiller les sauvegardes HBase Unités: minutes

Métriques de capacité de cluster

Les métriques suivantes indiquent les capacités actuelles ou cibles d'un cluster. Ces métriques sont disponibles uniquement lorsque le dimensionnement géré est activé. Pour les clusters composés de parcs d'instances, les métriques de capacité du cluster sont mesurées dans `Units`. Pour les clusters composés de groupes d'instances, les métriques de capacité du cluster sont mesurées dans `Nodes` ou `VCPU` en fonction du type d'unité utilisé dans la stratégie de mise à l'échelle gérée. Pour de plus amples informations, veuillez consulter Utilisation du dimensionnement géré par EMR dans Amazon EMR dans le Amazon EMR Guide de gestion.

Métrique	Description
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	Nombre total cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `TotalUnitsRunning` • `TotalNodesRunning` • `TotalVCPURunning`	Nombre total actuel d'unités/nœuds/vCPU disponibles dans un cluster en cours d'exécution. Lorsqu'un redimensionnement de cluster est demandé, cette métrique est mise à jour après l'ajout ou la suppression des nouvelles instances du cluster. Unités: Count
• `CoreUnitsRequested` • `CoreNodesRequested` • `CoreVCPURequested`	Nombre cible d'unités/nœuds/vCPU CORE dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `CoreUnitsRunning` • `CoreNodesRunning` • `CoreVCPURunning`	Nombre actuel d'unités/nœuds/vCPU CORE en cours d'exécution dans un cluster. Unités: Count
• `TaskUnitsRequested` • `TaskNodesRequested` • `TaskVCPURequested`	Nombre cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Nombre actuel d'unités/nœuds/vCPU TASK en cours d'exécution dans un cluster. Unités: Count

Dimensions des métriques Amazon EMR

Les données Amazon EMR peuvent être filtrées à l'aide des dimensions du tableau ci-dessous.

Dimension	Description
JobFlowId	Identique à cluster ID, qui est l'identifiant unique d'un cluster dans le formulaire `j-XXXXXXXXXXXXX`. Trouvez cette valeur en cliquant sur le bouton cluster dans la console Amazon EMR.
JobId	Identifiant d'une tâche au sein d'un cluster. Vous pouvez l'utiliser pour filtrer les métriques renvoyées par un cluster afin de n'afficher que celles qui s'appliquent à une tâche unique dans le cluster. Il prend la forme job_XXXXXXXXXXXX_XXXX.

View Cluster Application Metrics with Ganglia

Gris centraux sont disponibles avec Amazon EMR versions 4.2 et ultérieures. Ganglia est un projet open source qui est un système distribué évolutif conçu pour surveiller les clusters et les grilles tout en

minimisant l'impact sur leurs performances. Lorsque vous activez Ganglia sur votre cluster, vous pouvez générer des rapports et afficher la performance du cluster dans son ensemble, ainsi qu'inspecter la performance des instances de chaque nœud. Ganglia est également configuré pour intégrer et visualiser les métriques Hadoop et Spark. Pour plus d'informations, consultez Ganglia dans le Amazon EMR Guide de version.

Journalisation des appels d'API Amazon EMR dans AWS CloudTrail

Amazon EMR est intégré à AWS CloudTrail, un service qui fournit un enregistrement des actions réalisées par un utilisateur, un rôle ou un service AWS dans Amazon EMR. CloudTrail capture tous les appels d'API pour Amazon EMR en tant qu'événements. Les appels capturés incluent des appels de la console Amazon EMR et le code des appels vers les opérations d'API Amazon EMR. Si vous créez un journal de suivi, vous pouvez diffuser en continu les événements CloudTrail sur un compartiment Amazon S3, y compris les événements pour Amazon EMR. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à Amazon EMR, l'adresse IP source à partir de laquelle la demande a été effectuée, l'auteur de la demande et la date de la demande, ainsi que d'autres informations.

Pour en savoir plus sur CloudTrail, consultez AWS CloudTrail User Guide.

Informations Amazon EMR dans CloudTrail

CloudTrail est activé sur votre compte AWS lorsque vous créez le compte. Lorsqu'une activité a lieu dans Amazon EMR, cette activité est enregistrée dans un événement CloudTrail avec d'autres événements de services AWS dans Event history (Historique des événements). Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez Affichage des événements avec l'historique des événements CloudTrail.

Pour un enregistrement continu des événements dans votre compte AWS, y compris les événements pour Amazon EMR, créez un journal de suivi. Une journal de suivi permet à CloudTrail de livrer les fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal de suivi 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'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- Intégrations et services pris en charge par CloudTrail
- Configuration des Notifications de Amazon SNS pour CloudTrail
- Réception de fichiers journaux CloudTrail de plusieurs régions et Réception de fichiers journaux CloudTrail de plusieurs comptes

Toutes les actions Amazon EMR sont consignées par CloudTrail et sont documentées dans le Amazon EMR API Reference. Par exemple, les appels adressés aux actions `RunJobFlow`, `ListCluster` et `DescribeCluster` génèrent des entrées dans les fichiers journaux CloudTrail.

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification racine ou utilisateur AWS Identity and Access Management (IAM).

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

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

Exemple : entrées de fichier journal Amazon EMR

Un journal de suivi est une configuration qui active la livraison d'événements en tant que fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail contiennent une ou plusieurs entrées de journal. Un événement représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'action demandée, sur tous les paramètres, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne sont pas des séries ordonnées retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action RunJobFlow.

```
{
 "Records": [
 {
         "eventVersion":"1.01",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"EX_PRINCIPAL_ID",
            "arn":"arn:aws:iam::123456789012:user/temporary-user-xx-7M",
            "accountId":"123456789012",
            "userName":"temporary-user-xx-7M"
         },
         "eventTime":"2018-03-31T17:59:21Z",
         "eventSource":"elasticmapreduce.amazonaws.com",
         "eventName":"RunJobFlow",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"192.0.2.1",
         "userAgent":"aws-sdk-java/unknown-version Linux/xx Java_HotSpot(TM)_64-
Bit_Server_VM/xx",
         "requestParameters":{
            "tags":[
               {
                  "value":"prod",
                  "key":"domain"
               },
               {
                  "value":"us-west-2",
                  "key":"realm"
               },
               {
                  "value":"VERIFICATION",
                  "key":"executionType"
               }
            ],
            "instances":{
               "slaveInstanceType":"m5.xlarge",
               "ec2KeyName":"emr-integtest",
               "instanceCount":1,
               "masterInstanceType":"m5.xlarge",
               "keepJobFlowAliveWhenNoSteps":true,
               "terminationProtected":false
            },
            "visibleToAllUsers":false,
            "name":"MyCluster",
```

```
            "ReleaseLabel":"emr-5.16.0"
         },
         "responseElements":{
            "jobFlowId":"j-2WDJCGEG4E6AJ"
         },
         "requestID":"2f482daf-b8fe-11e3-89e7-75a3d0e071c5",
         "eventID":"b348a38d-f744-4097-8b2a-e68c9b424698"
      },
 ...additional entries
  ]
}
```

Connexion au cluster

Lorsque vous exécutez un cluster Amazon EMR, il vous suffit souvent d'exécuter une application pour analyser vos données, puis collecter les données de sortie à partir d'un compartiment Amazon S3. A d'autres moments, vous pouvez souhaiter interagir avec le nœud principal alors que le cluster est en cours d'exécution. Par exemple, vous pouvez souhaiter vous connecter au nœud principal pour exécuter des requêtes interactives, vérifier des fichiers journaux, résoudre un problème avec le cluster, surveiller les performances à l'aide d'une application comme Ganglia qui s'exécute sur le nœud principal, etc. Les sections suivantes décrivent les techniques que vous pouvez utiliser pour vous connecter au nœud principal.

Dans un cluster EMR, le nœud principal est une instance Amazon EC2 qui coordonne les instances EC2 qui s'exécutent en tant que nœuds de tâches et nœuds principaux. Le nœud principal expose un nom DNS public que vous pouvez utiliser pour vous y connecter. Par défaut, Amazon EMR crée des règles de groupe de sécurité pour le nœud principale et pour les nœuds principaux et de tâches, qui déterminent la façon dont vous accédez aux nœuds.

> **Note**
>
> Vous pouvez vous connecter au nœud principal uniquement pendant l'exécution d'un cluster. Lorsque le cluster s'arrête, l'instance EC2 qui agit en tant que nœud principal est mise hors service et n'est plus disponible. Pour vous connecter au nœud maître, vous devez également vous authentifier auprès du cluster. Vous pouvez soit utiliser Kerberos pour l'authentification, soit spécifier une clé privée de la paire de clés Amazon EC2 lorsque vous lancez le cluster. Pour plus d'informations sur la configuration de Kerberos, puis la connexion, consultez Use Kerberos Authentication (p. 255). Lorsque vous lancez un cluster à partir de la console, la clé privée de la paire de clés Amazon EC2 est spécifiée dans la section Sécurité et accès, sur la page Créer un cluster.

Par défaut, le groupe de sécurité ElasticMapReduce-master n'autorise pas l'accès SSH entrant. Vous pouvez avoir besoin d'ajouter une règle entrante qui autorise l'accès SSH (port TCP 22) à partir des sources pour lesquelles vous souhaitez bénéficier d'un accès. Pour plus d'informations sur la modification des règles du groupe de sécurité, consultez Ajout de règles au groupe de sécurité dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

> **Important**
>
> Ne modifiez pas les autres règles du groupe de sécurité ElasticMapReduce-master. La modification de ces règles peut interférer avec le fonctionnement du cluster.

Rubriques

- Connexion au nœud maître à l'aide de SSH (p. 368)
- Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372)

Connexion au nœud maître à l'aide de SSH

SSH (Secure Shell) est un protocole de réseau que vous pouvez utiliser pour créer une connexion sécurisée à un ordinateur distant. Après avoir établi une connexion, le terminal de votre ordinateur local se comporte comme s'il s'exécutait sur l'ordinateur distant. Les commandes que vous émettez localement s'exécutent sur l'ordinateur distant, et la sortie de commande de l'ordinateur distant s'affiche dans la fenêtre de votre terminal.

Lorsque vous utilisez SSH avec AWS, vous vous connectez à une instance EC2, c'est-à-dire à un serveur virtuel qui s'exécute dans le cloud. Lorsque vous travaillez avec Amazon EMR, l'utilisation la plus courante de SSH consiste à vous connecter à l'instance EC2 qui agit en tant que nœud principal du cluster.

Lorsque vous utilisez SSH pour vous connecter au nœud principal, vous pouvez surveiller le cluster et interagir avec lui. Vous pouvez émettre des commandes Linux sur le nœud maître, exécuter des applications telles que Hive et Pig de façon interactive, parcourir des annuaires, lire les fichiers journaux, et ainsi de suite. Vous pouvez également créer un tunnel dans votre connexion SSH pour afficher les interfaces web hébergées sur le nœud maître. Pour plus d'informations, consultez la section Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

Pour vous connecter au nœud maître à l'aide de SSH, vous avez besoin du nom DNS public du nœud maître. En outre, le groupe de sécurité associé au nœud principal doit avoir une règle entrante qui autorise le trafic SSH (port TCP 22) à partir d'une source qui inclut le client depuis lequel la connexion SSH provient. Vous aurez peut-être besoin d'ajouter une règle pour autoriser une connexion SSH à partir de votre client. Pour de plus amples informations sur la modification des règles du groupe de sécurité, veuillez consulter Contrôle du trafic réseau avec des groupes de sécurité (p. 301) et Ajout de règles au groupe de sécurité dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Récupération du nom DNS public du nœud maître

Vous pouvez récupérer le nom de serveur DNS public du nœud principal à l'aide de la console Amazon EMR et de l'AWS CLI.

Pour récupérer le nom DNS public du nœud principal à l'aide de la console Amazon EMR

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Dans la page Liste de clusters, sélectionnez le lien de votre cluster.
3. Notez la valeur Master public DNS (DNS public principal) qui s'affiche dans la partie supérieure de la page Cluster Details (Détails de cluster).

> **Note**
>
> Vous pouvez également choisir le lien SSH en regard du nom de serveur DNS public du nœud principal pour obtenir des informations sur la création d'une connexion SSH avec le nœud principal.

Pour récupérer le nom DNS public du nœud principal à l'aide de l'AWS CLI

1. Pour récupérer l'identifiant du cluster, tapez la commande suivante.

```
aws emr list-clusters
```

Vous obtenez la liste de vos clusters, y compris les ID des clusters. Notez l'ID du cluster auquel vous vous connectez.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "My cluster"
```

2. Pour afficher les instances de cluster, y compris le nom de serveur DNS public du nœud principal du cluster, tapez l'une des commandes suivantes. Remplacez *j-2AL4XXXXXX5T9* avec l'ID de cluster renvoyé par la commande précédente.

```
aws emr list-instances --cluster-id j-2AL4XXXXXX5T9
```

Ou :

```
aws emr describe-cluster --cluster-id j-2AL4XXXXXX5T9
```

Vous obtenez la liste des instances de cluster, y compris les noms DNS et les adresses IP. Notez la valeur de `PublicDnsName`.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040779.263,
        "CreationDateTime": 1408040515.535
    },
    "State": "RUNNING",
    "StateChangeReason": {}
},
"Ec2InstanceId": "i-e89b45e7",
```

```
"PublicDnsName": "ec2-###-##-##-###.us-west-2.compute.amazonaws.com"

"PrivateDnsName": "ip-###-##-##-###.us-west-2.compute.internal",
"PublicIpAddress": "##.###.###.##",
"Id": "ci-12XXXXXXXXFMH",
"PrivateIpAddress": "###.##.#.###"
```

Pour plus d'informations, consultez Commandes Amazon EMR dans l'AWS CLI.

Connexion au nœud maître à l'aide de SSH et d'une clé privée Amazon EC2 sous Linux, Unix et Mac OS X

Pour créer une connexion SSH authentifiée à l'aide d'un fichier de clé privée, vous devez spécifier la clé privée de la paire de clés Amazon EC2 lorsque vous lancez un cluster. Si vous lancez un cluster à partir de la console, la clé privée de la paire de clés Amazon EC2 est spécifiée dans la section Sécurité et accès, sur la page Créer un cluster. Pour plus d'informations sur l'accès à votre paire de clés, consultez Paires de clés Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Il est probable que votre ordinateur Linux comporte un client SSH par défaut. Par exemple, OpenSSH est installé sur la plupart des systèmes d'exploitation Linux, Unix et macOS. Vous pouvez vérifier un client SSH en tapant ssh dans la ligne de commande. Si votre ordinateur ne reconnaît pas la commande, installez un client SSH pour vous connecter au nœud principal. Le projet OpenSSH offre une implémentation gratuite de la suite entière des outils SSH. Pour de plus amples informations, veuillez consulter le site web OpenSSH.

Les instructions suivantes décrivent l'ouverture d'une connexion SSH sur le nœud maître Amazon EMR sous Linux, Unix et Mac OS X.

Pour configurer les autorisations sur les fichiers de clé privée de paire de clés

Avant de pouvoir utiliser votre clé privée de paire de clés Amazon EC2 pour créer une connexion SSH, vous devez définir des autorisations sur le fichier `.pem` afin que seul le propriétaire des clés soit autorisé à accéder au fichier. Cette action est obligatoire pour créer une connexion SSH à l'aide d'un terminal ou de l'AWS CLI.

1. Recherchez votre fichier `.pem`. Ces instructions supposent que le fichier est nommé `mykeypair.pem` et qu'il est stocké dans le répertoire de base de l'utilisateur actuel.
2. Pour définir les autorisations, saisissez la commande suivante. Remplacez *~/mykeypair.pem* avec l'emplacement et le nom de fichier de votre fichier de clé privée de paire de clés.

   ```
   chmod 400 ~/mykeypair.pem
   ```

   Si vous ne définissez pas d'autorisations sur le fichier `.pem`, vous recevez une erreur indiquant que votre fichier de clé n'est pas protégé et la clé sera rejetée. Pour vous connecter, il vous suffit de définir des autorisations sur le fichier de clé privée de paire de clés la première fois que vous l'utilisez.

Pour vous connecter au nœud maître à l'aide du terminal

1. Ouvrez une fenêtre du terminal. Sous Mac OS X, choisissez Applications > Utilities > Terminal (Applications > Services > Terminal). Sur d'autres distributions Linux, le terminal se trouve généralement sur Applications > Accessories > Terminal (Applications > Accessoires > Terminal).
2. Pour établir une connexion au nœud maître, tapez la commande suivante. Remplacez *ec2-###-##-##-###.compute-1.amazonaws.com* avec le nom DNS public principal de votre cluster et remplacez *~/mykeypair.pem* avec l'emplacement et le nom de fichier de votre `.pem` fichier.

```
ssh hadoop@ec2-###-##-##-###.compute-1.amazonaws.com -i ~/mykeypair.pem
```

> **Important**
>
> Vous devez utiliser le nom de connexion `hadoop` lorsque vous vous connectez au nœud principal Amazon EMR, sinon une erreur similaire à `Server refused our key` peut s'afficher.

3. Un avertissement indique que l'authenticité de l'hôte auquel vous vous connectez ne peut pas être vérifiée. Tapez `yes` pour continuer.
4. Lorsque vous avez terminé d'utiliser le nœud maître, tapez la commande suivante pour fermer la connexion SSH.

```
exit
```

Connexion au nœud maître à l'aide de l'AWS CLI

Vous pouvez créer une connexion SSH avec le nœud principal à l'aide de l'AWS CLI sous Windows et sous Linux, Unix et Mac OS X. Quelle que soit la plateforme, vous avez besoin du nom DNS public du nœud principal et de la clé privée de votre paire de clés Amazon EC2. Si vous utilisez l'AWS CLI sous Linux, Unix ou Mac OS X, vous devez également définir des autorisations sur le fichier de clé privée (`.pem` ou `.ppk`) comme indiqué dans Pour configurer les autorisations sur les fichiers de clé privée de paire de clés (p. 370).

Pour vous connecter au nœud maître à l'aide de l'AWS CLI

1. Pour récupérer l'identifiant du cluster, tapez :

```
aws emr list-clusters
```

   Vous obtenez la liste de vos clusters, y compris les ID des clusters. Notez l'ID du cluster auquel vous vous connectez.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

2. Tapez la commande suivante pour ouvrir une connexion SSH vers le nœud maître. Dans l'exemple suivant, remplacez *`j-2AL4XXXXXX5T9`* avec l'ID du cluster et remplacez *`~/mykeypair.key`* avec l'emplacement et le nom de fichier de votre `.pem` fichier (pour Linux, Unix et Mac OS X) ou `.ppk` fichier (pour Windows).

```
aws emr ssh --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

3. Lorsque vous avez terminé d'utiliser le nœud maître, fermez la fenêtre de l'AWS CLI.

Pour plus d'informations, consultez Commandes Amazon EMR dans l'AWS CLI.

Connexion au nœud maître à l'aide de SSH sous Windows

Les utilisateurs Windows peuvent utiliser un client SSH tel que PuTTY pour se connecter au nœud maître. Avant de vous connecter au nœud principal Amazon EMR, vous devez télécharger et installer PuTTY et PuTTYgen. Vous pouvez télécharger ces outils à partir de la page de téléchargement PuTTY.

PuTTY ne prend pas en charge de manière native le format de fichier de clé privée de paire de clés (`.pem`) généré par Amazon EC2. Vous utilisez PuTTYgen pour convertir votre fichier de clé au format PuTTY approprié (`.ppk`). Vous devez convertir votre clé dans ce format (`.ppk`) avant d'essayer de vous connecter au nœud principal à l'aide de PuTTY.

Pour plus d'informations sur la conversion de votre clé, consultez Conversion de votre clé privée à l'aide de PuTTYgen dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Pour vous connecter au nœud maître à l'aide de PuTTY

1. Ouvert `putty.exe`. Vous pouvez également lancer PuTTY à partir de la liste des programmes Windows.
2. Si nécessaire, dans la liste Category (Catégorie), choisissez Session.
3. Pour Nom d'hôte (ou adresse IP)type `hadoop@`*`MasterPublicDNS`*. Par exemple : `hadoop@`*`ec2-###-##-##-###.compute-1.amazonaws.com`*.
4. Dans la liste Category (Catégorie), sélectionnez Connection > SSH (Connexion > SSH), Auth.
5. Pour Private key file for authentication (Fichier de clé privée pour l'authentification), choisissez Browse (Parcourir), puis sélectionnez le fichier `.ppk` que vous avez généré.
6. Choisissez Open (Ouvrir) et Yes (Oui) pour ignorer l'alerte de sécurité PuTTY.

   Important

   Lorsque vous vous connectez au nœud maître, tapez `hadoop` si vous êtes invité à saisir un nom d'utilisateur.
7. Lorsque vous avez terminé d'utiliser le nœud maître, vous pouvez fermer la connexion SSH en fermant PuTTY.

   Note

   Pour éviter que la connexion SSH expire, vous pouvez choisir Connexion dans la liste Category (Catégorie) et sélectionner l'option Enable TCP_keepalives (Activer TCP_keepalives). Si vous disposez d'une session SSH active dans PuTTY, vous pouvez modifier vos paramètres en ouvrant le contexte (clic droit) pour la barre de titre PuTTY et en choisissant Modifier les paramètres.

Affichage des interfaces web hébergées sur des clusters Amazon EMR

Hadoop et les autres applications que vous installez sur votre cluster Amazon EMR publient des interfaces utilisateur en tant que sites web hébergés sur le nœud maître. Pour des raisons de sécurité, lors de l'utilisation des groupes de sécurité gérés par EMR, ces sites web sont uniquement disponibles sur le serveur web local du nœud maître et, par conséquent, vous devez vous connecter au nœud maître pour les afficher. Pour plus d'informations, consultez la section Connexion au nœud maître à l'aide de SSH (p. 368). Hadoop publie également les interfaces utilisateur en tant que sites web hébergés sur

les nœuds principaux et de tâches. Ces sites web sont également disponibles uniquement sur les serveurs web local sur les nœuds.

> **Warning**
>
> Il est possible de configurer un groupe de sécurité personnalisé pour autoriser l'accès entrant aux interfaces web. Gardez à l'esprit que tout port sur lequel vous autorisez le trafic entrant représente une faille de sécurité potentielle. Vérifiez attentivement les groupes de sécurité personnalisés pour vous assurer de réduire les failles de sécurité. Pour plus d'informations, consultez la section Contrôle du trafic réseau avec des groupes de sécurité (p. 301).

Le tableau suivant répertorie les interfaces web que vous pouvez afficher sur les instances de cluster. Ces interfaces Hadoop sont disponibles sur tous les clusters. Pour les interfaces d'instance principale, remplacez *master-public-dns-name* avec le DNS public maître listé sur le cluster Résumé dans la console EMR. Pour les interfaces d'instance de noyau et de tâche, remplacez *coretask-public-dns-name* avec le Nom DNS public répertorié pour l'instance. Pour rechercher un Public DNS name (Nom de DNS public), dans la console EMR, sélectionnez votre cluster dans la liste, choisissez l'onglet Hardware (Matériel), choisissez l' ID du groupe d'instances qui contient l'instance à laquelle vous souhaitez vous connecter, puis notez le Public DNS name (Nom de DNS public) répertorié pour l'instance.

> **Important**
>
> Pour accéder aux interfaces web, vous devez modifier les groupes de sécurité associés aux instances principales et de noyau afin qu'ils aient une règle entrante qui autorise le trafic SSH (port 22) à partir de clients approuvés, telles que l'adresse IP de votre ordinateur. Pour plus d'informations sur la modification des règles du groupe de sécurité, consultez Ajout de règles au groupe de sécurité dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Nom de l'interface	URI
Ganglia	http://*master-public-dns-name*/ganglia/
Hadoop HDFS NameNode (version EMR antérieure à 6.x)	https://*master-public-dns-name*: 50470/
Hadoop HDFS NameNode	https://*master-public-dns-name*: 50070/
Hadoop HDFS DataNode	https://*coretask-public-dns-name*: 50075/
Hadoop HDFS NameNode (EMR version 6.x)	https://*master-public-dns-name*: 9871/
Hadoop HDFS DataNode (version EMR antérieure à 6.x)	https://*coretask-public-dns-name*: 50475/
DataNode Hadoop HDFS (EMR version 6.x)	https://*coretask-public-dns-name*: 9865/
HBase	http://*master-public-dns-name*: 16010/
Hue	http://*master-public-dns-name*: 8888/
JupyterHub	https://*master-public-dns-name*: 9443/
Livy	http://*master-public-dns-name*: 8998/
Spark HistoryServer	http://*master-public-dns-name*: 18080/
Tez	http://*master-public-dns-name*: 8080/Interface de transmission

Nom de l'interface	URI
Gestionnaire de nœuds YARN	http://*coretask-public-dns-name*: 8042/
Gestionnaire de ressources YARN	http://*master-public-dns-name*: 8088/
Zeppelin	http://*master-public-dns-name*: 8890/

Etant donné que plusieurs interfaces spécifiques à l'application sont disponibles sur le nœud maître mais ne sont pas disponibles sur les nœuds principaux et de tâches, les instructions de ce document sont spécifiques au nœud maître Amazon EMR. Vous pouvez accéder aux interfaces web sur les nœuds principaux et de tâches de la même manière qu'aux interfaces web sur le nœud maître.

Il existe plusieurs façons d'accéder aux interfaces web sur le nœud maître. La méthode la plus simple et la plus rapide consiste à utiliser SSH pour vous connecter au nœud maître et à utiliser le navigateur texte Lynx afin d'afficher les sites web de votre client SSH. Toutefois, Lynx est un navigateur texte avec une interface utilisateur limitée qui ne peut pas afficher de graphiques. L'exemple suivant montre comment ouvrir l'interface Hadoop ResourceManager à l'aide de Lynx (les URL Lynx sont également fournies lorsque vous vous connectez au nœud maître à l'aide de SSH).

```
lynx http://ip-###-##-##-###.us-west-2.compute.internal:8088/
```

Il existe deux autres options pour accéder aux interfaces web sur le nœud maître, qui fournissent des fonctionnalités de navigateur complet. Choisissez l'une des options suivantes :

- Option 1 (recommandée pour les utilisateurs plus techniques) : Utilisez un client SSH pour vous connecter au nœud principal, configurez le tunnel SSH avec le transfert de port local et utilisez un navigateur Internet pour ouvrir les interfaces web hébergées sur le nœud principal. Cette méthode vous permet de configurer l'accès aux interfaces web sans utiliser de proxy SOCKS.
- Option 2 (recommandée pour les nouveaux utilisateurs) : Utilisez un client SSH pour vous connecter au nœud maître, configurez le tunnel SSH avec le transfert de port dynamique et configurez votre navigateur Internet pour utiliser un module complémentaire tel que FoxyProxy ou SwitchySharp pour gérer vos paramètres de proxy SOCKS. Cette méthode vous permet de filtrer automatiquement les URL en fonction des modèles de texte et de limiter les paramètres de proxy aux domaines qui correspondent à la forme du nom de DNS du nœud maître. Le module complémentaire du navigateur gère automatiquement l'activation et la désactivation du proxy lorsque vous basculez entre les sites web hébergés sur le nœud maître et ceux hébergés sur Internet. Pour plus d'informations sur la façon de configurer FoxyProxy pour Firefox et Google Chrome, consultez Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître (p. 378).

Avec Amazon EMR version 5.25.0 ou ultérieure, vous pouvez accéder à l'interface utilisateur du serveur d'historique Spark à partir de la console sans configurer un proxy web via une connexion SSH. Pour plus d'informations, consultez Accès en un clic au serveur d'historique Spark permanent.

Rubriques

- Option 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port local (p. 375)
- Option 2, Partie 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port dynamique (p. 375)
- Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître (p. 378)
- Accès aux interfaces web sur le nœud maître à l'aide de la console (p. 380)

Option 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port local

Pour vous connecter au serveur Web local sur le nœud maître, vous créez un tunnel SSH entre votre ordinateur et le nœud maître. Cette action est également appelée réacheminement de port. Si vous ne souhaitez pas utiliser un proxy SOCKS, vous pouvez configurer un tunnel SSH vers le nœud maître à l'aide du réacheminement de port. Avec le réacheminement de port local, vous spécifiez les ports locaux qui sont inutilisés pour transférer le trafic vers des ports à distance spécifiques sur le serveur web local du nœud maître.

La configuration d'un tunnel SSH à l'aide du réacheminement de port local nécessite le nom de serveur DNS public du nœud maître et le fichier de clé privée de votre paire de clés. Pour plus d'informations sur la façon de rechercher le nom de serveur DNS public du nœud principal, consultez Pour récupérer le nom DNS public du nœud principal à l'aide de la console Amazon EMR (p. 368). Pour plus d'informations sur l'accès à votre paire de clés, consultez Paires de clés Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Pour plus d'informations sur les sites que vous pouvez afficher sur le nœud maître, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

Configuration d'un tunnel SSH vers le nœud maître à l'aide du réacheminement de port local sous Linux, Unix et Mac OS X

Pour configurer un tunnel SSH à l'aide du réacheminement de port local dans le terminal

1. Ouvrez une fenêtre du terminal. Sous Mac OS X, choisissez Applications > Utilities > Terminal (Applications > Services > Terminal). Sur d'autres distributions Linux, le terminal se trouve généralement sur Applications > Accessories > Terminal (Applications > Accessoires > Terminal).
2. Tapez la commande suivante pour ouvrir un tunnel SSH sur votre ordinateur local. Cette commande accède à l'interface web ResourceManager en réacheminant le trafic du port local 8157 (port local inutilisé choisi de façon aléatoire) vers le port 8088 sur le serveur web local du nœud maître. Dans la commande , remplacez `~/mykeypair.pem` avec l'emplacement et le nom de fichier de votre `.pem` et remplacer `ec2-###-##-##-###.compute-1.amazonaws.com` avec le nom DNS public principal de votre cluster.

   ```
   ssh -i ~/mykeypair.pem -N -L 8157:ec2-###-##-##-###.compute-1.amazonaws.com:8088
    hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   Lorsque cette commande est émise, le terminal reste ouvert et ne retourne pas de réponse.

   > **Note**
   >
   > `-L` indique l'utilisation du réacheminement de port local, qui vous permet de spécifier un port local utilisé pour transférer des données au port à distance identifié sur le serveur web local du nœud maître.

3. Pour ouvrir l'interface Web ResourceManager dans votre navigateur, tapez : `http://localhost:8157/` dans la barre d'adresse.
4. Lorsque vous avez terminé d'utiliser les interfaces web sur le nœud maître, fermez la fenêtre du terminal.

Option 2, Partie 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port dynamique

Pour vous connecter au serveur Web local sur le nœud maître, vous créez un tunnel SSH entre votre ordinateur et le nœud maître. Cette action est également appelée réacheminement de port. Si vous créez

votre tunnel SSH à l'aide du réacheminement de port dynamique, l'ensemble du trafic acheminé vers un port local inutilisé spécifié est réacheminé vers le serveur web local sur le nœud maître. Cette action crée un proxy SOCKS. Vous pouvez ensuite configurer votre navigateur Internet afin qu'il utilise un module complémentaire comme FoxyProxy ou SwitchySharp pour gérer vos paramètres de proxy SOCKS. A l'aide d'un module complémentaire de gestion de proxy, vous pouvez filtrer automatiquement les URL en fonction de modèles de texte et limiter les paramètres de proxy aux domaines qui correspondent à la forme du nom de serveur DNS public du nœud maître. Le module complémentaire du navigateur gère automatiquement l'activation et la désactivation du proxy lorsque vous basculez entre les sites web hébergés sur le nœud maître et ceux hébergés sur Internet.

Avant de commencer, vous devez connaître le nom de serveur DNS public du nœud maître et le fichier de clé privée de votre paire de clés. Pour plus d'informations sur la façon de rechercher le nom de serveur DNS public du nœud principal, consultez Pour récupérer le nom DNS public du nœud principal à l'aide de la console Amazon EMR (p. 368). Pour plus d'informations sur l'accès à votre paire de clés, consultez Paires de clés Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux. Pour plus d'informations sur les sites que vous pouvez afficher sur le nœud maître, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

Configuration d'un tunnel SSH vers le nœud maître à l'aide du réacheminement de port dynamique sous Linux, Unix et Mac OS X

Pour configurer un tunnel SSH à l'aide du réacheminement de port dynamique sous Linux, Unix et Mac OS X

1. Ouvrez une fenêtre du terminal. Sous Mac OS X, choisissez Applications > Utilities > Terminal (Applications > Services > Terminal). Sur d'autres distributions Linux, le terminal se trouve généralement sur Applications > Accessories > Terminal (Applications > Accessoires > Terminal).
2. Tapez la commande suivante pour ouvrir un tunnel SSH sur votre ordinateur local. Remplacez *~/mykeypair.pem* avec l'emplacement et le nom de fichier de votre `.pem` fichier, remplacer *8157* avec un numéro de port local inutilisé, et remplacez *c2-###-##-##-###.compute-1.amazonaws.com* avec le nom DNS public principal de votre cluster.

   ```
   ssh -i ~/mykeypair.pem -N -D 8157 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   Lorsque cette commande est émise, le terminal reste ouvert et ne retourne pas de réponse.

   > **Note**
   >
   > `-D` indique l'utilisation du réacheminement de port dynamique, qui vous permet de spécifier un port local utilisé pour transférer des données à tous les ports à distance sur le serveur web local du nœud maître. Le réacheminement de port dynamique crée un proxy SOCKS local qui écoute sur le port spécifié dans la commande.
3. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez la section Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître (p. 378).
4. Lorsque vous avez terminé d'utiliser les interfaces web sur le nœud maître, fermez la fenêtre du terminal.

Configuration d'un tunnel SSH à l'aide du réacheminement de port dynamique avec l'AWS CLI

Vous pouvez créer une connexion SSH avec le nœud principal à l'aide de l'AWS CLI sous Windows et sous Linux, Unix et Mac OS X. Si vous utilisez l'AWS CLI sous Linux, Unix ou Mac OS X, vous devez définir les autorisations sur le fichier `.pem` comme indiqué dans Pour configurer les autorisations sur les fichiers de clé privée de paire de clés (p. 370). Si vous utilisez le AWS CLI sur Windows, PuTTY doit apparaître dans

la variable d'environnement de chemin d'accès ou vous pouvez recevoir une erreur telle que OpenSSH or PuTTY not available.

Pour configurer un tunnel SSH à l'aide du réacheminement de port dynamique avec l'AWS CLI

1. Créez une connexion SSH avec le nœud maître, comme illustré dans Connexion au nœud maître à l'aide de l'AWS CLI (p. 371).
2. Pour récupérer l'identifiant du cluster, tapez :

```
aws emr list-clusters
```

   Vous obtenez la liste de vos clusters, y compris les ID des clusters. Notez l'ID du cluster auquel vous vous connectez.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

3. Tapez la commande suivante pour ouvrir un tunnel SSH vers le nœud maître à l'aide du réacheminement de port dynamique. Dans l'exemple suivant, remplacez *j-2AL4XXXXXX5T9* avec l'ID du cluster et remplacez *~/mykeypair.key* avec l'emplacement et le nom de fichier de votre `.pem` fichier (pour Linux, Unix et Mac OS X) ou `.ppk` fichier (pour Windows).

```
aws emr socks --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

   Note

   La commande SOCKS configure automatiquement le réacheminement de port dynamique sur le port local 8157. Actuellement, ce paramètre ne peut pas être modifié.

4. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez la section Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître (p. 378).
5. Lorsque vous avez terminé d'utiliser les interfaces web sur le nœud principal, fermez la fenêtre AWS CLI.

   Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Configuration d'un tunnel SSH vers le nœud maître à l'aide du réacheminement de port dynamique sous Windows

Les utilisateurs Windows peuvent utiliser un client SSH tel que PuTTY pour créer un tunnel SSH pour le nœud maître. Avant de vous connecter au nœud principal Amazon EMR, vous devez télécharger et installer PuTTY et PuTTYgen. Vous pouvez télécharger ces outils à partir de la page de téléchargement PuTTY.

PuTTY ne prend pas en charge de manière native le format de fichier de clé privée de paire de clés (`.pem`) généré par Amazon EC2. Vous utilisez PuTTYgen pour convertir votre fichier de clé au format PuTTY approprié (`.ppk`). Vous devez convertir votre clé dans ce format (`.ppk`) avant d'essayer de vous connecter au nœud principal à l'aide de PuTTY.

Pour plus d'informations sur la conversion de votre clé, consultez Conversion de votre clé privée à l'aide de PuTTYgen dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Pour configurer un tunnel SSH à l'aide du réacheminement de port dynamique sous Windows

1. Double-cliquez sur `putty.exe` pour lancer PuTTY. Vous pouvez également lancer PuTTY à partir de la liste des programmes Windows.

   Note

   Si vous avez déjà une session SSH active avec le nœud principal, vous pouvez ajouter un tunnel en cliquant avec le bouton droit sur la barre de titre PuTTY et en choisissant Change Settings (Modifier les paramètres).
2. Si nécessaire, dans la liste Category (Catégorie), choisissez Session.
3. Dans le Nom d'hôte champ, type **`hadoop@`***`MasterPublicDNS`*. Par exemple : **`hadoop@`***`ec2-###-##-##-###.compute-1.amazonaws.com`*.
4. Dans la liste Category (Catégorie), développez Connection > SSH (Connexion > SSH), puis choisissez Auth.
5. Pour Private key file for authentication (Fichier de clé privée pour l'authentification), choisissez Browse (Parcourir), puis sélectionnez le fichier `.ppk` que vous avez généré.

   Note

   PuTTY ne prend pas en charge de manière native le format de fichier de clé privée de paire de clés (`.pem`) généré par Amazon EC2. Vous utilisez PuTTYgen pour convertir votre fichier de clé au format PuTTY approprié (`.ppk`). Vous devez convertir votre clé dans ce format (`.ppk`) avant d'essayer de vous connecter au nœud principal à l'aide de PuTTY.
6. Dans la liste Category (Catégorie), développez Connection > SSH (Connexion > SSH), puis choisissez Tunnels.
7. Dans le champ Source port (Port source), saisissez `8157` (un port local inutilisé).
8. Laissez le champ Destination vide.
9. Sélectionnez les options Dynamic (Dynamique) et Auto.
10. Choisissez Add (Ajouter) et Open (Ouvrir).
11. Choisissez Yes (Oui) pour ignorer l'alerte de sécurité PuTTY.

    Important

    Lorsque vous vous connectez au nœud maître, tapez `hadoop` si vous êtes invité à saisir un nom d'utilisateur.
12. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez la section Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître (p. 378).
13. Lorsque vous avez terminé d'utiliser les interfaces web sur le nœud maître, fermez la fenêtre PuTTY.

Option 2, Partie 2 : Configurer les paramètres de proxy pour afficher les sites web hébergés sur le nœud maître

Si vous utilisez un tunnel SSH avec réacheminement de port dynamique, vous devez utiliser un module complémentaire de gestion de proxy SOCKS pour contrôler les paramètres de proxy dans votre navigateur.

A l'aide d'un outil de gestion de proxy SOCKS, vous pouvez filtrer automatiquement les URL en fonction de modèles de texte et limiter les paramètres de proxy aux domaines qui correspondent à la forme du nom de serveur DNS public du nœud maître. Le module complémentaire du navigateur gère automatiquement l'activation et la désactivation du proxy lorsque vous basculez entre les sites web hébergés sur le nœud maître et ceux hébergés sur Internet. Pour gérer vos paramètres de proxy, configurez votre navigateur afin qu'il utilise un module complémentaire comme FoxyProxy ou SwitchySharp.

Pour plus d'informations sur la création d'un tunnel SSH, consultez Option 2, Partie 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port dynamique (p. 375). Pour plus d'informations sur les interfaces web disponibles, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

L'exemple suivant montre une configuration FoxyProxy Google Chrome. Les paramètres appropriés qui sont chargés depuis le fichier de configuration dans l'exemple sont les suivants :

- Hôte ou adresse IP : défini sur localhost avec le port 8157 dans l'exemple. Vous devez définir ce port avec le numéro de port local que vous avez utilisé pour établir le tunnel SSH avec le nœud maître dans Option 2, Partie 1 : Configuration d'un tunnel SSH vers le nœud maître à l'aide du transfert de port dynamique (p. 375). Ce port doit aussi correspondre au numéro de port que vous utilisez dans PuTTY ou à un autre émulateur de terminal que vous utilisez pour vous connecter.
- La configuration de SOCKS v5 est spécifiée.
- Les informations d'identification de connexion ne sont pas spécifiées.
- Modèles URL

  Les modèles URL suivants sont sur liste blanche et spécifiés par un type de modèle de caractère générique :

  - Les modèles *ec2*.amazonaws.com* et *10*.amazonaws.com* correspondent au nom DNS public des clusters dans les régions des USA.
  - Les modèles *ec2*.compute* et *10*.compute* correspondent au nom DNS public des clusters de toutes les autres régions.
  - Le modèle 10. offre un accès aux fichiers journaux de JobTracker dans Hadoop. Modifiez ce filtre s'il est en conflit avec votre plan d'accès réseau.

Configuration de FoxyProxy pour Google Chrome

Vous pouvez configurer FoxyProxy pour Google Chrome, Mozilla Firefox et Microsoft Internet Explorer. FoxyProxy fournit un ensemble d'outils de gestion de proxy qui vous permettent d'utiliser un serveur proxy pour les URL qui correspondent à des modèles correspondant aux domaines utilisés par les instances Amazon EC2 dans votre cluster Amazon EMR.

Installer et configurer FoxyProxy à l'aide de Google Chrome

1. Consultez https://chrome.google.com/webstore/search/foxy%20proxy, et suivez les liens et instructions pour ajouter FoxyProxy à Chrome.
2. A l'aide d'un éditeur de texte, créez un fichier nommé `foxyproxy-settings.xml` avec le contenu suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<foxyproxy>
   <proxies>
      <proxy name="emr-socks-proxy" id="2322596116" notes="" fromSubscription="false"
 enabled="true" mode="manual" selectedTabIndex="2" lastresort="true"
 animatedIcons="true" includeInCycle="true" color="#0055E5" proxyDNS="true"
 noInternalIPs="false" autoconfMode="pac" clearCacheBeforeUse="false"
 disableCache="false" clearCookiesBeforeUse="false" rejectCookies="false">
         <matches>
```

```
            <match enabled="true" name="*ec2*.amazonaws.com*"
 pattern="*ec2*.amazonaws.com*" isRegEx="false" isBlackList="false" isMultiLine="false"
 caseSensitive="false" fromSubscription="false" />
            <match enabled="true" name="*ec2*.compute*" pattern="*ec2*.compute*"
 isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="false"
 fromSubscription="false" />
            <match enabled="true" name="10.*" pattern="http://10.*"
 isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="false"
 fromSubscription="false" />
            <match enabled="true" name="*10*.amazonaws.com*"
 pattern="*10*.amazonaws.com*" isRegEx="false" isBlackList="false" isMultiLine="false"
 caseSensitive="false" fromSubscription="false" />
            <match enabled="true" name="*10*.compute*" pattern="*10*.compute*"
 isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="false"
 fromSubscription="false" />
            <match enabled="true" name="*.compute.internal*"
 pattern="*.compute.internal*" isRegEx="false" isBlackList="false" isMultiLine="false"
 caseSensitive="false" fromSubscription="false"/>
            <match enabled="true" name="*.ec2.internal* " pattern="*.ec2.internal*"
 isRegEx="false" isBlackList="false" isMultiLine="false" caseSensitive="false"
 fromSubscription="false"/>
    </matches>
         <manualconf host="localhost" port="8157" socksversion="5" isSocks="true"
 username="" password="" domain="" />
      </proxy>
   </proxies>
</foxyproxy>
```

3. Gérez les extensions dans Chrome (accédez à chrome://extensions).
4. Choisissez Options pour FoxyProxy Standard.
5. Sur la page FoxyProxy , choisissez Import/Export.
6. Sur la page Import/Export, choisissez Choisir un fichier, accédez à l'emplacement du fichier `foxyproxy-settings.xml` que vous avez créé, sélectionnez le fichier, puis choisissez Ouvrir.
7. Choisissez Replace (Remplacer) lorsque vous êtes invité à remplacer les paramètres existants.
8. Pour Proxy mode (Mode de proxy), choisissez Use proxies based on their predefined patterns and priorities (Utiliser les proxies selon leurs modèles et priorités prédéfinis).
9. Pour ouvrir les interfaces Web, dans la barre d'adresse de votre navigateur, tapez *`master-public-dns`* suivi du numéro de port ou de l'URL.

   Pour obtenir une liste complète des interfaces web sur le nœud maître, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

Accès aux interfaces web sur le nœud maître à l'aide de la console

Si vous avez déjà un tunnel SSH configuré avec le nœud maître Amazon EMR et utilisant le réacheminement de port dynamique, vous pouvez ouvrir les interfaces web à l'aide de la console.

Pour ouvrir les interfaces web à l'aide de la console

1. Vérifiez que vous avez établi un tunnel SSH avec le nœud maître et que vous disposez d'un module complémentaire de gestion de proxy configuré pour votre navigateur.
2. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
3. Dans la page Liste de clusters, choisissez le lien relatif à votre cluster.
4. Dans les détails de cluster, pour Connexions, cliquez sur le lien de l'interface web que vous souhaitez ouvrir dans votre navigateur.

5. Sinon, choisissez le lien Tout afficher pour afficher les liens vers toutes les interfaces web disponibles sur le nœud principal de votre cluster. Lorsque vous utilisez les liens, les interfaces s'ouvrent dans votre navigateur.

Si vous n'avez pas de tunnel SSH ouvert avec le nœud maître, choisissez Activer la connexion Web pour obtenir des instructions sur la création d'un tunnel.

Note

Si vous disposez d'un tunnel SSH configuré et utilisant le réacheminement de port local, la console Amazon EMR ne détecte pas la connexion.

Arrêter un cluster

Cette section décrit les méthodes d'arrêter un cluster. Pour plus d'informations sur l'activation de la protection de la résiliation et l'arrêt automatique des clusters, consultez Control Cluster Termination (p. 110). Vous pouvez arrêter des clusters dans les états `STARTING`, `RUNNING` ou `WAITING`. Un cluster dans l'état `WAITING` doit être arrêté ou il s'exécute indéfiniment, générant des frais sur votre compte. Vous pouvez arrêter un cluster qui n'est pas parvenu à quitter l'état `STARTING` ou ne peut pas effectuer une étape.

Si vous résiliez un cluster sur lequel une protection de la résiliation est définie, vous devez tout d'abord désactiver la protection de la résiliation avant de pouvoir résilier le cluster. Les clusters peuvent être résiliés à l'aide de la console, de l'AWS CLI ou par programmation à l'aide de l'API `TerminateJobFlows`.

En fonction de la configuration du cluster, il peut falloir de 5 à 20 minutes au cluster pour s'arrêter totalement et libérer les ressources allouées, telles que des instances EC2.

Mettre fin à un cluster à l'aide de la console

Vous pouvez mettre fin à une ou plusieurs clusters à l'aide de la console Amazon EMR. Les étapes d'arrêt d'un cluster dans la console varient selon si la protection de la résiliation est activée ou non. Pour arrêter un cluster protégé, vous devez tout d'abord désactiver la protection de la résiliation.

Pour arrêter un cluster avec la protection de la résiliation désactivée

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Sélectionnez le cluster à arrêter. Vous pouvez sélectionner plusieurs clusters et les suspendre simultanément.
3. Choisissez Terminer.
4. A l'invite, choisissez Résilier.

   Amazon EMR résilie les instances dans le cluster et arrête d'enregistrer des données de journal.

Pour arrêter un cluster avec la protection de la résiliation activée

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/.
2. Sur la page Liste de clusters, sélectionnez le cluster à arrêter. Vous pouvez sélectionner plusieurs clusters et les suspendre simultanément.
3. Choisissez Terminer.
4. Lorsque vous y êtes invité, choisissez Modification pour désactiver la protection de la résiliation. Si vous avez sélectionné plusieurs clusters, choisissez Tout désactiver pour désactiver simultanément la protection de la résiliation pour tous les clusters.
5. Dans la boîte de dialogue Résilier les clusters, pour Protection de la résiliation, choisissez Désactivé, puis cliquez sur la case à cocher pour confirmer.
6. Cliquez sur Résilier.

   Amazon EMR résilie les instances dans le cluster et arrête d'enregistrer des données de journal.

Arrêter un cluster à l'aide de l'AWS CLI

Pour arrêter un cluster non protégé à l'aide de l'AWS CLI

Pour résilier un cluster non protégé à l'aide de l'AWS CLI, utilisez la sous-commande `terminate-clusters` avec le paramètre --cluster-ids.

- Saisissez la commande suivante pour arrêter un seul cluster et remplacez *j-3KVXXXXXXX7UG* par l'ID de votre cluster.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
```

  Pour arrêter plusieurs clusters, saisissez la commande suivante et remplacez *j-3KVXXXXXXX7UG* et *j-WJ2XXXXXX8EU* par vos ID de cluster.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
```

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Pour arrêter un cluster protégé à l'aide de l'AWS CLI

Pour arrêter un cluster protégé à l'aide de l'AWS CLI, commencez par désactiver la protection de la résiliation à l'aide de la sous-commande `modify-cluster-attributes` avec le paramètre `--no-termination-protected`. Utilisez ensuite la sous-commande `terminate-clusters` avec le paramètre `--cluster-ids` pour l'arrêter.

1. Saisissez la commande suivante pour désactiver la protection de la résiliation et remplacez *j-3KVTXXXXXX7UG* avec votre ID de cluster.

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-
protected
```

2. Pour arrêter le cluster, saisissez la commande suivante et remplacez *j-3KVXXXXXXX7UG* par l'ID de votre cluster.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
```

  Pour arrêter plusieurs clusters, saisissez la commande suivante et remplacez *j-3KVXXXXXXX7UG* et *j-WJ2XXXXXX8EU* par vos ID de cluster.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
```

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Arrêter un cluster à l'aide de l'API

L'opération `TerminateJobFlows` termine le traitement de l'étape, télécharge toutes données de journal d'Amazon EC2 vers Amazon S3 (si configuré) et résilie le cluster Hadoop. Un cluster s'arrête également automatiquement si vous définissez `KeepJobAliveWhenNoSteps` sur `False` dans une demande `RunJobFlows`.

Vous pouvez utiliser cette action pour arrêter un cluster unique ou une liste de clusters par leurs ID de cluster.

Pour de plus amples 'informations sur les paramètres d'entrée uniques pour `TerminateJobFlows`, veuillez consulter TerminateJobFlows. Pour de plus amples informations sur les paramètres génériques dans la demande, veuillez consulter Paramètres de demande communs.

Dimensionnement des ressources de cluster

Vous pouvez ajuster automatiquement ou manuellement le nombre d'instances Amazon EC2 disponibles dans un cluster Amazon EMR en fonction de charges de travail aux exigences diverses. Pour utiliser le dimensionnement automatique, vous avez deux options. Vous pouvez activer le dimensionnement géré par EMR ou créer une stratégie de dimensionnement automatique personnalisé. Le tableau suivant décrit les différences entre les deux options.

	Dimensionnement géré par EMR	Dimensionnement automatique personnalisé
Stratégies et règles de dimensionnement	Aucune stratégie n'est requise. EMR gère l'activité de dimensionnement automatique en évaluant en permanence les métriques de cluster et en prenant des décisions optimisées en matière de dimensionnement.	Vous devez définir et gérer les stratégies et règles de dimensionnement automatique, telles que les conditions spécifiques qui déclenchent les activités de dimensionnement, les périodes d'évaluation, les temps de stabilisation, etc.
Versions EMR prises en charge	Amazon EMR version 5.30.0 et ultérieure (sauf Amazon EMR version 6.0.0)	Amazon EMR version 4.0.0 et ultérieure
Composition de cluster prise en charge	Groupes d'instances ou parcs d'instances	Groupes d'instances uniquement
Configuration des limites de dimensionnement	Les limites de dimensionnement sont configurées pour l'ensemble du cluster.	Les limites de dimensionnement ne peuvent être configurées que pour chaque groupe d'instances.
Fréquence de l'évaluation des métriques	Toutes les 5 à 10 secondes Une évaluation plus fréquente des métriques permet à EMR de prendre des décisions de dimensionnement plus précises.	Vous ne pouvez définir les périodes d'évaluation que par incréments de cinq minutes.
Applications prises en charge	Seules les applications YARN sont prises en charge, telles que Spark, Hadoop, Hive, Flink. Les autres applications, telles que Presto, ne sont actuellement pas prises en charge.	Vous pouvez choisir les applications prises en charge lors de la définition des règles de dimensionnement automatique.

Considérations

- Un cluster Amazon EMR se compose toujours d'un ou de trois nœuds principaux. Vous ne pouvez pas dimensionner le nombre de nœuds principaux après la configuration initiale du cluster. Vous pouvez dimensionner seulement les nœuds principaux et les nœuds de tâches dans un cluster.
- La reconfiguration et le redimensionnement d'un groupe d'instances ne peuvent pas s'effectuer en même temps. Si une reconfiguration est initiée pendant le redimensionnement d'un groupe d'instances, la reconfiguration ne peut pas démarrer tant que le redimensionnement du groupe d'instances n'est pas terminé, et inversement.

Rubriques

- Utilisation du dimensionnement automatique géré par EMR dans Amazon EMR (p. 385)
- Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401)
- Redimensionnement manuel d'un cluster en cours d'exécution (p. 410)
- Diminution de capacité des clusters (p. 416)

Utilisation du dimensionnement automatique géré par EMR dans Amazon EMR

avec Amazon EMR versions 5.30.0 et ultérieures (sauf pour Amazon EMR 6.0.0), vous pouvez activer le dimensionnement géré par EMR pour augmenter ou diminuer automatiquement le nombre d'instances ou d'unités dans votre cluster en fonction de la charge de travail. EMR évalue en permanence les métriques de cluster pour prendre des décisions de dimensionnement qui optimisent vos clusters en termes de coût et de vitesse. Cette fonction est disponible pour les clusters composés de groupes d'instances ou de parcs d'instances.

Vous devez configurer les paramètres suivants pour le dimensionnement géré. La limite s'applique uniquement aux nœuds principaux et aux nœuds de tâches. Le nœud principal ne peut pas être dimensionné après la configuration initiale.

- Minimum (`MinimumCapacityUnits`) – Limite inférieure de la capacité EC2 autorisée dans un cluster. Elle est mesurée par le biais de cœurs ou d'instances d'unités de traitement centrales virtuelles (vCPU) pour les groupes d'instances. Elle est mesurée par unités pour les parcs d'instances.
- Maximum (`MaximumCapacityUnits`) – La limite supérieure de la capacité EC2 autorisée dans un cluster. Elle est mesurée par le biais de cœurs ou d'instances d'unités de traitement centrales virtuelles (vCPU) pour les groupes d'instances. Elle est mesurée par unités pour les parcs d'instances.
- Limite à la demande (`MaximumOnDemandCapacityUnits`) (Facultatif) – La limite supérieure de la capacité EC2 autorisée pour le type de marché à la demande dans un cluster. Si ce paramètre n'est pas spécifié, il est défini par défaut sur la valeur de `MaximumCapacityUnits`.

  Ce paramètre est utilisé pour répartir l'allocation de capacité entre les instances à la demande et les instances Spot. Par exemple, si vous définissez le paramètre minimum comme 2 instances, le paramètre maximum comme 100 instances, la limite à la demande comme 10 instances, puis la mise à l'échelle gérée par EMR dimensionne jusqu'à 10 instances à la demande et alloue la capacité restante aux instances Spot. Pour plus d'informations, consultez la section Scénarios d'allocation de nœuds (p. 387).
- Nœuds principaux maximum (`MaximumCoreCapacityUnits`) (Facultatif) – Limite supérieure de capacité EC2 autorisée pour le type de nœud principal dans un cluster. Si ce paramètre n'est pas spécifié, il est défini par défaut sur la valeur de `MaximumCapacityUnits`.

  Ce paramètre est utilisé pour répartir l'allocation de capacité entre les nœuds principaux et de tâches. Par exemple, si vous définissez le paramètre minimum comme 2 instances, le maximum comme 100 instances, le nœud principal maximum comme 17 instances, le dimensionnement géré par EMR

dimensionne jusqu'à 17 nœuds principaux et alloue les 83 instances restantes aux nœuds de tâches. Pour plus d'informations, consultez la section Scénarios d'allocation de nœuds (p. 387).

Pour plus d'informations sur les paramètres de dimensionnement géré, consultez `ComputeLimits`.

Considérations et restrictions

- Le dimensionnement géré par EMR est actuellement disponible dans 22 régions AWS : Est des États-Unis (Virginie du Nord et Ohio), Ouest des États-Unis (Oregon et Californie du Nord), Amérique du Sud (Sao Paulo), Europe (Francfort, Irlande, Londres, Milan, Paris et Stockholm), Canada (Centre), Asie-Pacifique (Hong Kong, Bombay, Séoul, Singapour, Sydney et Tokyo), Moyen-Orient (Bahreïn), Afrique (Cap), Chine (Pékin) exploitée par Sinnet, Chine et NCD.
- Le dimensionnement géré par EMR fonctionne uniquement avec les applications YARN, telles que Spark, Hadoop, Hive, Flink. Actuellement, elle ne prend pas en charge les applications non basées sur YARN, telles que Presto.
- Lorsque vous définissez le nœud principal maximal et la limite à la demande pour le dimensionnement géré par EMR, tenez compte des différences entre les groupes d'instances et les parcs d'instances. Chaque groupe d'instances se compose du même type d'instance et de la même option d'achat pour les instances : À la demande ou Spot. Pour chaque parc d'instances, vous pouvez spécifier jusqu'à cinq types d'instance, qui peuvent être provisionnés en tant qu'instances à la demande et instances Spot. Pour plus d'informations, consultez Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes, Options de parc d'instances, et Scénarios d'allocation de nœuds (p. 387).

Rubriques

- Présentation de la stratégie et des scénarios d'allocation de nœuds (p. 386)
- Présentation des métriques de dimensionnement géré (p. 390)
- Utilisation de l'AWS Management Console pour configurer le dimensionnement géré (p. 394)
- Options de console mises à jour pour le dimensionnement du cluster (p. 396)
- Utilisation de l'AWS CLI pour configurer le dimensionnement géré (p. 397)
- Utilisation d'AWS SDK for Java pour la configuration du dimensionnement géré (p. 399)

Présentation de la stratégie et des scénarios d'allocation de nœuds

Cette section donne une vue d'ensemble de la stratégie d'allocation de nœuds et des scénarios de dimensionnement courants que vous pouvez utiliser avec le dimensionnement géré par EMR.

Stratégie d'allocation de nœuds

La mise à l'échelle gérée par EMR alloue les nœuds principaux et de tâches en fonction des stratégies d'augmentation et de réduction suivantes :

Stratégie de mise à l'échelle

- Le dimensionnement géré par EMR ajoute d'abord la capacité aux nœuds principaux, puis aux nœuds de tâches jusqu'à ce que la capacité maximale autorisée soit atteinte ou jusqu'à ce que la capacité cible de mise à l'échelle souhaitée soit atteinte.
- Si le `MaximumCoreCapacityUnits` est défini, puis Amazon EMR met à l'échelle les nœuds principaux jusqu'à ce que les unités principales atteignent la limite maximale autorisée. Toute la capacité restante est ajoutée aux nœuds de tâches.
- Si le `MaximumOnDemandCapacityUnits` est défini, puis Amazon EMR met à l'échelle le cluster en utilisant les instances à la demande jusqu'à ce que les unités à la demande atteignent la limite maximale autorisée. Toute la capacité restante est ajoutée à l'aide des instances Spot.

- Si les deux `MaximumCoreCapacityUnits` et `MaximumOnDemandCapacityUnits` les paramètres sont définis, Amazon EMR prend en compte les deux limites pendant la mise à l'échelle.

  Par exemple, si le `MaximumCoreCapacityUnits` est inférieur à `MaximumOnDemandCapacityUnits`, EMR dimensionne d'abord les nœuds principaux jusqu'à ce que la limite de capacité principale soit atteinte. Pour la capacité restante, EMR utilise d'abord les instances à la demande pour dimensionner les nœuds de tâche jusqu'à ce que la limite à la demande soit atteinte, puis utilise Instances Spot pour les nœuds de tâches.

Stratégie de réduction

- La mise à l'échelle gérée par EMR supprime d'abord les nœuds de tâche, puis les nœuds principaux jusqu'à ce que la capacité cible de réduction souhaitée soit atteinte. Le cluster ne se met jamais à l'échelle en dessous des contraintes minimales dans la stratégie de dimensionnement gérée.
- Au sein de chaque type de nœud (nœuds principaux ou nœuds de tâches), le dimensionnement géré par EMR supprime d'abord les instances Spot, puis les instances à la demande.

Si le cluster n'a pas de charge, alors Amazon EMR annule l'ajout de nouvelles instances à partir d'une évaluation précédente et effectue des opérations de diminution en charge. Si le cluster a une charge lourde, Amazon EMR annule la suppression des instances et effectue des opérations de mise à l'échelle.

Considérations relatives à l'allocation de nœuds

Nous vous recommandons d'utiliser l'option d'achat à la demande pour les nœuds principaux afin d'éviter la perte de données HDFS en cas de récupération Spot. Vous pouvez utiliser l'option d'achat Spot pour les nœuds de tâche afin de réduire les coûts et d'obtenir une exécution de tâche plus rapide lorsque d'autres instances Spot sont ajoutées aux nœuds de tâche.

Scénarios d'allocation de nœuds

Vous pouvez créer différents scénarios de mise à l'échelle en fonction de vos besoins en configurant les paramètres de nœud principal Maximum, Minimum, Limite à la demande et Maximum dans différentes combinaisons.

Scénario 1: Nœuds de noyau de mise à l'échelle uniquement

Pour dimensionner les nœuds principaux uniquement, les paramètres de dimensionnement gérés doivent répondre aux exigences suivantes :

- La limite à la demande est égale à la limite maximale.
- Le nœud principal maximum est égal à la limite maximum.

Lorsque la limite à la demande et les paramètres de nœud de noyau maximum ne sont pas spécifiés, les deux paramètres par défaut sont la limite maximale.

Les exemples suivants illustrent le scénario de dimensionnement des nœuds principaux uniquement.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Groupes d'instances Noyau : 1 à la demande	`UnitType`: Instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20	Mettez à l'échelle entre 1 et 20 instances ou unités de parc d'instances sur les

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Tâche : 1 à la demande et 1 à la demande	`MaximumOnDemandCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 20	nœuds principaux à l'aide du type à la demande. Pas de mise à l'échelle sur les nœuds de tâche.
Parcs d'instances Noyau : 1 à la demande Tâche : 1 à la demande et 1 à la demande	Type d'unité : Unités de parc d'instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 20	

Scénario 2: Nœuds de tâche de mise à l'échelle uniquement

Pour dimensionner les nœuds de tâche uniquement, les paramètres de dimensionnement gérés doivent répondre à l'exigence suivante :

- Le nœud principal maximum doit être égal à la limite minimum.

Les exemples suivants illustrent le scénario de dimensionnement des nœuds de tâches uniquement.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Groupes d'instances Noyau : 2 À la demande Tâche : 1 point	`UnitType`: Instances `MinimumCapacityUnits`: 2 `MaximumCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 2	Maintenez les nœuds principaux stables à 2 et mettez uniquement à l'échelle les nœuds de tâches entre 0 et 18 instances ou unités de parc d'instances. La capacité entre les limites minimales et maximales est ajoutée aux nœuds de tâche uniquement.
Parcs d'instances Noyau : 2 À la demande Tâche : 1 point	`UnitType`: Unités de parc d'instances `MinimumCapacityUnits`: 2 `MaximumCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 2	

Scénario 3 : Uniquement les instances à la demande dans le cluster

Pour avoir uniquement des instances à la demande, votre cluster et les paramètres de dimensionnement géré doivent répondre à l'exigence suivante :

- La limite à la demande est égale à la limite maximale.

  Lorsque la limite à la demande n'est pas spécifiée, la valeur par défaut du paramètre est la limite maximale. La valeur par défaut indique que Amazon EMR dimensionne les instances à la demande uniquement.

Si le nœud principal maximum est inférieur à la limite maximale, le paramètre de nœud principal maximum peut être utilisé pour répartir l'allocation de capacité entre les nœuds principaux et les nœuds de tâches.

Pour activer ce scénario dans un cluster composé de groupes d'instances, tous les groupes de nœuds du cluster doivent utiliser le type de marché à la demande lors de la configuration initiale.

Les exemples suivants illustrent le scénario d'avoir des instances à la demande dans l'ensemble du cluster.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Groupes d'instances Noyau : 1 à la demande Tâche : 1 à la demande	`UnitType`: Instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 12	Mettez à l'échelle entre 1 et 12 instances ou unités de parc d'instances sur les nœuds principaux à l'aide du type à la demande. Mettez à l'échelle la capacité restante à l'aide de à la demande sur les nœuds de tâches. Aucune mise à l'échelle à l'aide d'instances Spot.
Parcs d'instances Noyau : 1 à la demande Tâche : 1 à la demande	`UnitType`: Unités de parc d'instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 20 `MaximumCoreCapacityUnits`: 12	

4e scénario : Uniquement les instances Spot dans le cluster

Pour avoir uniquement des instances Spot, les paramètres de dimensionnement gérés doivent répondre à l'exigence suivante :

- La limite à la demande est définie sur 0.

Si le nœud principal maximum est inférieur à la limite maximale, le paramètre de nœud principal maximum peut être utilisé pour répartir l'allocation de capacité entre les nœuds principaux et les nœuds de tâches.

Pour activer ce scénario dans un cluster composé de groupes d'instances, le groupe d'instances principal doit utiliser l' Point option d'achat lors de la configuration initiale. S'il n'y a pas d'instance Spot dans le groupe d'instances de tâche, le dimensionnement géré par EMR crée un groupe de tâches à l'aide des instances Spot si nécessaire.

Les exemples suivants illustrent le scénario d'avoir des instances Spot dans l'ensemble du cluster.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Groupes d'instances Noyau : 1 point Tâche : 1 point	`UnitType`: Instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 0 USD	Mettez à l'échelle entre 1 et 20 instances ou unités de parc d'instances sur les nœuds

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Parcs d'instances Noyau : 1 point Tâche : 1 point	`UnitType`: Unités de parc d'instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 0 USD	principaux à l'aide de l'instance Spot. Pas de mise à l'échelle à l'aide du type à la demande.

Scénario 5 : Mise à l'échelle des instances à la demande sur les nœuds principaux et des instances Spot sur les nœuds de tâches

Pour dimensionner des instances à la demande sur des nœuds principaux et des instances Spot sur des nœuds de tâche, les paramètres de dimensionnement gérés doivent répondre aux exigences suivantes :

- La limite à la demande doit être égale au nœud principal maximum.
- La limite à la demande et le nœud principal maximal doivent être inférieurs à la limite maximale.

Pour activer ce scénario dans un cluster composé de groupes d'instances, le groupe de nœuds principal doit utiliser l'option d'achat à la demande.

Les exemples suivants illustrent le scénario de dimensionnement des instances à la demande sur les nœuds principaux et des instances Spot sur les nœuds de tâches.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle
Groupes d'instances Noyau : 1 à la demande Tâche : 1 à la demande et 1 à la demande	`UnitType`: Instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 7 `MaximumCoreCapacityUnits`: 7	Mettez à l'échelle jusqu'à 6 unités à la demande sur le nœud principal car il existe déjà 1 unité à la demande sur le nœud de tâche et la limite maximale pour On-Demand est 7. Puis dimensionnez jusqu'à 13 unités Spot sur les nœuds de tâche.
Parcs d'instances Noyau : 1 à la demande Tâche : 1 à la demande et 1 à la demande	`UnitType`: Unités de parc d'instances `MinimumCapacityUnits`: 1 `MaximumCapacityUnits`: 20 `MaximumOnDemandCapacityUnits`: 7 `MaximumCoreCapacityUnits`: 7	

Présentation des métriques de dimensionnement géré

Amazon EMR publie des métriques haute résolution avec des données à une granularité d'une minute lorsque le dimensionnement géré est activé pour un cluster. Vous pouvez afficher les événements sur chaque début et fin de nouvelle définition de taille contrôlé par le dimensionnement géré à l'aide de la console Amazon EMR ou de la console Amazon CloudWatch. Pour de plus amples informations, veuillez consulter Supervision des événements CloudWatch dans le Amazon EMR Guide de gestion.

Les métriques suivantes indiquent les capacités actuelles ou cibles d'un cluster. Ces métriques sont disponibles uniquement lorsque le dimensionnement géré est activé. Pour les clusters composés de parcs d'instances, les métriques de capacité du cluster sont mesurées dans `Units`. Pour les clusters composés de groupes d'instances, les métriques de capacité du cluster sont mesurées dans `Nodes` ou `vCPU` en fonction du type d'unité utilisé dans la stratégie de mise à l'échelle gérée.

Métrique	Description
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	Nombre total cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `TotalUnitsRunning` • `TotalNodesRunning` • `TotalVCPURunning`	Nombre total actuel d'unités/nœuds/vCPU disponibles dans un cluster en cours d'exécution. Lorsqu'un redimensionnement de cluster est demandé, cette métrique est mise à jour après l'ajout ou la suppression des nouvelles instances du cluster. Unités: Count
• `CoreUnitsRequested` • `CoreNodesRequested` • `CoreVCPURequested`	Nombre cible d'unités/nœuds/vCPU CORE dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `CoreUnitsRunning` • `CoreNodesRunning` • `CoreVCPURunning`	Nombre actuel d'unités/nœuds/vCPU CORE en cours d'exécution dans un cluster. Unités: Count
• `TaskUnitsRequested` • `TaskNodesRequested` • `TaskVCPURequested`	Nombre cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités: Count
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Nombre actuel d'unités/nœuds/vCPU TASK en cours d'exécution dans un cluster. Unités: Count

Les métriques suivantes indiquent l'état d'utilisation du cluster et des applications. Ces métriques sont disponibles pour tous Amazon EMR fonctions , mais sont publiées à une résolution plus élevée avec des données à une granularité d'une minute lorsque la mise à l'échelle gérée est activée pour un cluster. Vous pouvez mettre en corrélation les métriques suivantes avec les métriques de capacité de cluster du tableau précédent pour comprendre les décisions de dimensionnement géré.

Métrique	Description
`AppsCompleted`	Nombre de demandes soumises à YARN ayant été traitées. Cas d'utilisation : Moniteur cluster progression Unités: Count
`AppsPending`	Nombre d'applications soumises à YARN qui se trouvent dans un état d'attente. Cas d'utilisation : Moniteur cluster progression

Métrique	Description
	Unités: Count
`AppsRunning`	Nombre d'applications soumises à YARN qui sont en cours d'exécution. Cas d'utilisation : Moniteur cluster progression Unités: Count
`ContainerAllocated`	Nombre de conteneurs de ressources attribués par ResourceManager. Cas d'utilisation : Moniteur cluster progression Unités: Count
`ContainerPending`	Nombre de conteneurs dans la file d'attente qui n'ont pas encore été alloués. Cas d'utilisation : Moniteur cluster progression Unités: Count
`ContainerPendingRatio`	Le rapport entre les conteneurs en attente et les conteneurs attribués (Rapport de conteneurs en attente = Conteneurs en attente/Conteneurs attribués). Si les Conteneurs attribués = 0, le Rapport des conteneurs en attente = Conteneurs en attente. La valeur du Rapport des conteneurs en attente est exprimée sous forme de nombre et non de pourcentage. Cette valeur est utile pour dimensionner les ressources de cluster en fonction du comportement d'attribution des conteneurs. Unités: Count
`HDFSUtilizatioin`	Pourcentage de stockage HDFS actuellement utilisé. Cas d'utilisation : Analyser cluster performance Unités: Pourcentage
`IsIdle`	Indique qu'un cluster ne s'exécute plus, mais est encore en actif et génère des frais. Il est défini sur 1 si aucune tâche ni aucun travail n'est en cours d'exécution, et défini sur 0 dans le cas contraire. Cette valeur est vérifiée à intervalles de cinq minutes et une valeur de 1 indique uniquement que le cluster a été inactif lors de la vérification, et non pas qu'il a été inactif pendant les cinq minutes entières. Pour éviter les fausses erreurs, vous devez déclencher une alarme lorsque cette valeur est 1 pendant plusieurs contrôles consécutifs de 5 minutes. Par exemple, vous pouvez déclencher une alarme pour cette valeur si elle renvoie 1 pendant au moins 30 minutes. Cas d'utilisation : Moniteur cluster performance Unités: Booléen

Métrique	Description
`MemoryAvailableMB`	Quantité de mémoire disponible à allouer. Cas d'utilisation : Moniteur cluster progression Unités: Count
`MRActiveNodes`	Nombre de nœuds exécutant actuellement des tâches ou travaux MapReduce. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfActiveNodes`. Cas d'utilisation : Moniteur cluster progression Unités: Count
`YARNMemoryAvailablePercentage`	Le pourcentage de mémoire disponible restante pour YARN (Pourcentage de mémoire disponible YARN = Mémoire disponible en Mo/Total de mémoire en Mo). Cette valeur est utile pour dimensionner les ressources de cluster en fonction de l'utilisation de mémoire YARN. Unités: Pourcentage

Représentation graphique des métriques de dimensionnement géré

Vous pouvez créer une représentation graphique des métriques pour visualiser les modèles de charge de travail de votre cluster et les décisions de dimensionnement correspondantes prises par le dimensionnement géré par EMR, comme présenté dans les étapes suivantes.

Pour créer une représentation graphique des métriques de dimensionnement gérés dans la console CloudWatch

1. Ouvrez la console CloudWatch.
2. Dans le panneau de navigation, choisissez Amazon EMR. Vous pouvez rechercher par l'identifiant de cluster du cluster à surveiller.
3. Faites défiler jusqu'à la métrique que vous souhaitez représenter graphiquement. Ouvrez une métrique pour afficher le graphique.
4. Pour représenter graphiquement une ou plusieurs métriques, cochez la case en regard de chaque métrique.

L'exemple suivant illustre l'activité de dimensionnement géré par EMR d'un cluster. Le graphique présente trois périodes de dimensionnement (diminution) automatique, qui permettent de réaliser des économies lorsque la charge de travail est moins active.

Toutes les métriques de capacité et d'utilisation du cluster sont publiées à intervalles d'une minute. Des informations statistiques supplémentaires sont également associées à chaque donnée d'une minute, ce qui vous permet de tracer diverses fonctions telles que `Percentiles`, `Min`, `Max`, `Sum`, `Average`, `SampleCount`.

Par exemple, le graphique suivant trace la même métrique `YARNMemoryAvailablePercentage` à différents percentiles, P10, P50, P90, P99, ainsi que `Sum`, `Average`, `Min`, `SampleCount`.

Utilisation de l'AWS Management Console pour configurer le dimensionnement géré

Lorsque vous créez un cluster, vous pouvez configurer le dimensionnement géré à l'aide de l'option rapide ou des options de configuration de cluster avancées. Vous pouvez également créer ou modifier une stratégie de dimensionnement géré pour un cluster en cours d'exécution en modifiant les paramètres Managed Scaling (Dimensionnement géré) sur la page Summary (Récapitulatif) ou Hardware (Matériel).

Pour utiliser l'option rapide afin de configurer le dimensionnement géré lors de la création d'un cluster

1. Ouvrez la console Amazon EMR, choisissez Créer un cluster et ouvrez Créer un cluster - Options rapides.
2. Dans la section Configuration du matériel, dans le champ Cluster scaling (Dimensionnement du cluster) sélectionnez Use EMR-managed scaling) (Utiliser le dimensionnement géré par EMR).

3. Spécifiez le nombre minimum et maximum d'instances principales et de tâches.

Pour utiliser l'option avancée afin de configurer le dimensionnement géré lors de la création d'un cluster

1. Dans le Amazon EMR de la console, sélectionnez Créer un cluster, sélectionnez Aller aux options avancées, choisissez des options pour Étape 1 : Logiciels et étapes, puis allez à Étape 2 : Configuration matérielle.
2. Dans la section Cluster composition (Composition du cluster), sélectionnez Parcs d'instances ou Groupes d'instances uniformes.
3. Dans la section Cluster scaling (Dimensionnement du cluster), sélectionnez Enable cluster scaling (Activer le dimensionnement du cluster). Sélectionnez ensuite Use EMR managed scaling (Utiliser le dimensionnement géré par EMR). Puis spécifiez le Minimum et Maximum nombre d'instances ou d'unités de parc d'instances et Limite à la demande.

Pour les clusters composés de groupes d'instances, vous pouvez également choisir Create a custom automatic scaling policy (Créer une stratégie de dimensionnement automatique personnalisée) si vous souhaitez définir des stratégies de dimensionnement automatique personnalisées pour chaque groupe

d'instances. Pour plus d'informations, consultez la section Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401).

Pour modifier un cluster existant

1. Ouvrez la console Amazon EMR, sélectionnez votre cluster dans la liste des clusters, puis développez la section Matériel.
2. Dans la section Cluster scaling (Dimensionnement du cluster), sélectionnez Edit (Modifier) pour le dimensionnement géré par EMR.

3. Dans le Mise à l'échelle du cluster , spécifiez de nouvelles valeurs pour le Minimum et Maximum nombre d'instances et le Limite à la demande.

Options de console mises à jour pour le dimensionnement du cluster

Les options de dimensionnement du cluster dans la console ont été modifiées. Les options sont regroupées dans une section Cluster Scaling (Dimensionnement du cluster) sur la page Configuration du matériel de Créer un cluster - Options avancées. Les options de dimensionnement du cluster peuvent également être modifiées sous l'onglet Matériel d'un cluster en cours d'exécution.

Vous pouvez sélectionner Use EMR-managed scaling (Utiliser le dimensionnement géré par EMR) ou Create a custom automatic scaling policy (Créer une stratégie de dimensionnement automatique personnalisée) lorsque vous activez le dimensionnement du cluster. Use EMR-managed scaling (Utiliser le dimensionnement géréepar EMR) est disponible uniquement avec Amazon EMR version 5.30.0 et ultérieures. Pour plus d'informations, consultez la section Dimensionnement des ressources de cluster (p. 384).

Les captures d'écran suivantes présentent les différences entre le dimensionnement automatique précédent et les options de dimensionnement du cluster actuelles dans la console.

des modifications sur le Configuration matérielle page

Lorsque vous utilisez l'option avancée pour créer un cluster, les options de dimensionnement automatique sont déplacées du tableau des groupes d'instances vers la section Cluster scaling (Dimensionnement du cluster) de la page Configuration du matériel .

Modifications dans l'onglet Matériel

Dans l'onglet Matériel d'un cluster en cours d'exécution, les options de dimensionnement automatique sont déplacées vers le tableau Cluster scaling (Dimensionnement du cluster) sur la même page.

Utilisation de l'AWS CLI pour configurer le dimensionnement géré

Vous pouvez utiliser des commandes de l'AWS CLI pour demander à Amazon EMR de configurer le dimensionnement géré lorsque vous créez un cluster. Vous pouvez utiliser une syntaxe raccourcie, en spécifiant la configuration JSON compatible avec les commandes adéquates, ou vous pouvez indiquer un fichier contenant la configuration JSON. Vous pouvez également appliquer une stratégie

de dimensionnement géré à un cluster existant et supprimer une stratégie de dimensionnement géré précédemment appliquée. En outre, vous pouvez récupérer les détails de configuration d'une stratégie de dimensionnement à partir d'un cluster en cours d'exécution.

Activation du dimensionnement géré pendant le lancement du cluster

Vous pouvez activer le dimensionnement géré pendant le lancement du cluster, comme le montre l'exemple suivant.

```
aws emr create-cluster \
 --service-role EMR_DefaultRole \
 --release-label emr-5.30.1 \
 --name EMR_Managed_Scaling_Enabled_Cluster \
 --applications Name=Spark Name=Hbase \
 --ec2-attributes KeyName=keyName,InstanceProfile=EMR_EC2_DefaultRole \
 --instance-groups InstanceType=m4.xlarge,InstanceGroupType=MASTER,InstanceCount=1
 InstanceType=m4.xlarge,InstanceGroupType=CORE,InstanceCount=2 \
 --region us-east-1 \
 --managed-scaling-policy
 ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

Vous pouvez également spécifier une configuration de stratégie gérée à l'aide de l'option --managed-scaling-policy lorsque vous utilisez `create-cluster`.

Application d'une stratégie de dimensionnement géré à un cluster existant

Vous pouvez appliquer une stratégie de dimensionnement géré à un cluster existant, comme le montre l'exemple suivant.

```
aws emr put-managed-scaling-policy
--cluster-id j-123456
--managed-scaling-policy ComputeLimits='{MinimumCapacityUnits=1,
MaximumCapacityUnits=10,  MaximumOnDemandCapacityUnits=10, UnitType=Instances}'
```

Vous pouvez également appliquer une stratégie de dimensionnement géré à un cluster existant à l'aide de la commande `aws emr put-managed-scaling-policy`. L'exemple suivant utilise une référence à un fichier JSON, `managedscaleconfig.json`, qui spécifie la configuration de la stratégie de dimensionnement géré.

```
aws emr put-managed-scaling-policy --cluster-id j-123456 --managed-scaling-policy file://./
managedscaleconfig.json
```

L'exemple suivant présente le contenu du fichier `managedscaleconfig.json`, qui définit la stratégie de dimensionnement géré.

```
{
    "ComputeLimits": {
        "UnitType": "Instances",
        "MinimumCapacityUnits": 1,
        "MaximumCapacityUnits": 10,
        "MaximumOnDemandCapacityUnits": 10
    }
}
```

Récupération d'une configuration de stratégie de dimensionnement géré

La commande `GetManagedScalingPolicy` récupère la configuration de la stratégie. Par exemple, la commande suivante extrait la configuration pour le cluster avec un ID de cluster de `j-123456`.

```
aws emr get-managed-scaling-policy --cluster-id j-123456
```

La commande produit l'exemple de résultat suivant.

```
{
   "ManagedScalingPolicy": {
      "ComputeLimits": {
         "MinimumCapacityUnits": 1,
         "MaximumOnDemandCapacityUnits": 10,
         "MaximumCapacityUnits": 10,
         "UnitType": "Instances"
      }
   }
}
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Suppression d'une stratégie de dimensionnement géré

La commande `RemoveManagedScalingPolicy` supprime la configuration de stratégie. Par exemple, la commande suivante supprime la configuration pour le cluster dont l'ID est `j-123456`.

```
aws emr remove-managed-scaling-policy --cluster-id j-123456
```

Utilisation d'AWS SDK for Java pour la configuration du dimensionnement géré

L'extrait de programme suivant montre comment configurer le dimensionnement géré avec AWS SDK for Java :

```
package com.amazonaws.emr.sample;

import java.util.ArrayList;
import java.util.List;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.Application;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimits;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimitsUnitType;
import com.amazonaws.services.elasticmapreduce.model.InstanceGroupConfig;
import com.amazonaws.services.elasticmapreduce.model.JobFlowInstancesConfig;
import com.amazonaws.services.elasticmapreduce.model.ManagedScalingPolicy;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowRequest;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowResult;

public class CreateClusterWithManagedScalingWithIG {

 public static void main(String[] args) {
  AWSCredentials credentialsFromProfile = getCreadentials("AWS-Profile-Name-Here");
```

```
 /**
  * Create an EMR client using the credentials and region specified in order to create the
cluster
  */
 AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
  .withCredentials(new AWSStaticCredentialsProvider(credentialsFromProfile))
  .withRegion(Regions.US_EAST_1)
  .build();

 /**
  * Create Instance Groups - Master, Core, Task
  */
 InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
   .withInstanceCount(1)
   .withInstanceRole("MASTER")
   .withInstanceType("m4.large")
   .withMarket("ON_DEMAND");

 InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
  .withInstanceCount(4)
  .withInstanceRole("CORE")
  .withInstanceType("m4.large")
  .withMarket("ON_DEMAND");

 InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
  .withInstanceCount(5)
  .withInstanceRole("TASK")
  .withInstanceType("m4.large")
  .withMarket("ON_DEMAND");

 List<InstanceGroupConfig> igConfigs = new ArrayList<>();
 igConfigs.add(instanceGroupConfigMaster);
 igConfigs.add(instanceGroupConfigCore);
 igConfigs.add(instanceGroupConfigTask);

       /**
        *  specify applications to be installed and configured when EMR creates the
cluster
        */
 Application hive = new Application().withName("Hive");
 Application spark = new Application().withName("Spark");
 Application ganglia = new Application().withName("Ganglia");
 Application zeppelin = new Application().withName("Zeppelin");

 /**
  * Managed Scaling Configuration -
        * Using UnitType=Instances for clusters composed of instance groups
  *
        * Other options are:
        * UnitType = VCPU ( for clusters composed of instance groups)
        * UnitType = InstanceFleetUnits ( for clusters composed of instance fleets)
        **/
 ComputeLimits computeLimits = new ComputeLimits()
   .withMinimumCapacityUnits(1)
   .withMaximumCapacityUnits(20)
   .withUnitType(ComputeLimitsUnitType.Instances);

 ManagedScalingPolicy managedScalingPolicy = new ManagedScalingPolicy();
 managedScalingPolicy.setComputeLimits(computeLimits);

 // create the cluster with a managed scaling policy
 RunJobFlowRequest request = new RunJobFlowRequest()
         .withName("EMR_Managed_Scaling_TestCluster")
         .withReleaseLabel("emr-5.30.1")          // Specifies the EMR release version
label, we recommend the latest release
         .withApplications(hive,spark,ganglia,zeppelin)
```

```
         .withLogUri("s3://path/to/my/emr/logs")  // A URI in S3 for log files is required
when debugging is enabled.
         .withServiceRole("EMR_DefaultRole")      // If you use a custom IAM service role,
replace the default role with the custom role.
         .withJobFlowRole("EMR_EC2_DefaultRole")  // If you use a custom EMR role for EC2
instance profile, replace the default role with the custom EMR role.
         .withInstances(new JobFlowInstancesConfig().withInstanceGroups(igConfigs)
             .withEc2SubnetId("subnet-123456789012345")
             .withEc2KeyName("my-ec2-key-name")
             .withKeepJobFlowAliveWhenNoSteps(true))
         .withManagedScalingPolicy(managedScalingPolicy);
   RunJobFlowResult result = emr.runJobFlow(request);

   System.out.println("The cluster ID is " + result.toString());
}

public static AWSCredentials getCreadentials(String profileName) {
 // specifies any named profile in .aws/credentials as the credentials provider
 try {
  return new ProfileCredentialsProvider("AWS-Profile-Name-Here")
    .getCredentials();
       } catch (Exception e) {
           throw new AmazonClientException(
                   "Cannot load credentials from .aws/credentials file. " +
                   "Make sure that the credentials file exists and that the profile name
is defined within it.",
                   e);
       }
}

public CreateClusterWithManagedScalingWithIG() { }
}
```

Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances

Le dimensionnement automatique avec une stratégie personnalisée dans les versions 4.0 et ultérieures d'Amazon EMR vous permet d'augmenter ou de diminuer le nombre de nœuds principaux et de nœuds de tâches dans un cluster en fonction d'une métrique CloudWatch et d'autres paramètres spécifiés dans une stratégie de dimensionnement. Le dimensionnement automatique avec une stratégie personnalisée est disponible avec la configuration des groupes d'instances. Il ne l'est pas lorsque vous utilisez des parcs d'instances. Pour plus d'informations sur les groupes d'instances et sur les parcs d'instances, consultez Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes (p. 148).

> **Note**
>
> Pour utiliser le dimensionnement automatique avec une fonction de stratégie personnalisée dans Amazon EMR, vous devez définir `true` pour le paramètre `VisibleToAllUsers` lorsque vous créez un cluster. Pour plus d'informations, consultez SetVisibleToAllUsers.

La stratégie de dimensionnement fait partie de la configuration du groupe d'instances. Vous pouvez spécifier une stratégie lors de la configuration initiale d'un groupe d'instances, ou en modifiant un groupe d'instances dans un cluster existant, même quand ce groupe d'instances est actif. Chaque groupe d'instances dans un cluster, à l'exception du groupe d'instances maître, peut avoir sa propre stratégie de dimensionnement, qui se compose de règles de dimensionnement d'augmentation et de diminution. Les règles de dimensionnement (augmentation et diminution) peuvent être configurées indépendamment, avec des paramètres différents pour chaque règle.

Vous pouvez configurer des stratégies de dimensionnement à l'aide de la AWS Management Console de l'AWS CLI ou de l'API Amazon EMR. Lorsque vous utilisez l'AWS CLI ou l'API Amazon EMR, vous spécifiez la stratégie de dimensionnement au format JSON. En outre, lorsque vous utilisez l'AWS CLI ou

l'API Amazon EMR, vous pouvez spécifier des métriques CloudWatch personnalisées. Les métriques personnalisées ne sont pas disponibles pour la sélection à l'aide de la AWS Management Console. Lorsque vous créez initialement une stratégie de dimensionnement à l'aide de la console, une stratégie par défaut appropriée pour de nombreuses applications est préconfigurée pour vous aider à démarrer. Vous pouvez supprimer ou modifier les règles par défaut.

Même si le dimensionnement automatique vous permet d'ajuster la capacité d'un cluster EMR à la volée, vous devez toujours prendre en compte les exigences de charge de travail de base et planifier les configurations du groupe d'instances et du groupe de nœuds. Pour plus d'informations, consultez Consignes pour la configuration de cluster.

> Note
>
> Pour la plupart des charges de travail, il est conseillé de configurer les règles de dimensionnement (diminution et augmentation) pour optimiser l'utilisation des ressources. La configuration d'une seule de ces deux règles implique le redimensionnement manuel du nombre d'instances après une activité de dimensionnement. En d'autres termes, cela met en place une stratégie « unidirectionnelle » d'augmentation ou de diminution avec une réinitialisation manuelle.

Création du rôle IAM pour le dimensionnement automatique

Le dimensionnement automatique dans Amazon EMR nécessite un rôle IAM disposant des autorisations pour ajouter ou mettre hors service des instances lorsque des activités de dimensionnement sont lancées. Un rôle par défaut configuré avec la stratégie d'approbation et la stratégie de rôle appropriées, `EMR_AutoScaling_DefaultRole`, est disponible à cette fin. Lorsque vous créez un cluster avec une stratégie de dimensionnement à l'aide de la AWS Management Console pour la première fois, Amazon EMR crée le rôle par défaut et associe la stratégie gérée par défaut pour les autorisations, `AmazonElasticMapReduceforAutoScalingRole`.

Lorsque vous créez un cluster avec une stratégie de dimensionnement automatique à l'aide de l'AWS CLI, vous devez d'abord vous assurer que le rôle IAM par défaut existe, ou que vous avez un rôle IAM avec une stratégie attachée qui fournit les autorisations appropriées. Pour créer le rôle par défaut, vous pouvez exécuter la commande `create-default-roles` avant de créer un cluster. Vous pouvez alors spécifier l'option `--auto-scaling-role EMR_AutoScaling_DefaultRole` lorsque vous créez un cluster. Sinon, vous pouvez créer un rôle de dimensionnement automatique personnalisé, puis le spécifier lorsque vous créez un cluster, par exemple `--auto-scaling-role` *`MyEMRAutoScalingRole`*. Si vous créez un rôle de dimensionnement automatique personnalisé pour Amazon EMR, nous vous recommandons de baser les stratégies d'autorisations pour votre rôle personnalisé en fonction de la stratégie gérée. Pour plus d'informations, consultez la section Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).

Présentation des règles de dimensionnement automatique

Lorsqu'une règle d'augmentation déclenche une activité de dimensionnement pour un groupe d'instances, les instances Amazon EC2 sont ajoutées au groupe d'instances selon vos règles. Des applications comme Apache Spark, Apache Hive et Presto peuvent utiliser de nouveaux nœuds dès que l'instance Amazon EC2 passe à l'état `InService`. Vous pouvez également configurer une règle de dimensionnement (diminution) qui met des instances hors service et supprime des nœuds. Pour plus d'informations sur le cycle de vie des instances Amazon EC2 dont le dimensionnement est automatique, consultez Cycle de vie de dimensionnement automatique dans le Amazon EC2 Auto Scaling Guide de l'utilisateur.

Vous pouvez configurer la façon dont un cluster met des instances Amazon EC2 hors service. Pour la facturation, vous pouvez choisir de mettre hors service à l'échéance horaire de l'instance Amazon EC2 ou lorsque la tâche est terminée. Ce paramètre s'applique aux opérations de dimensionnement automatique et de redimensionnement manuel. Pour en savoir plus sur cette configuration, consultez Diminution de capacité des clusters (p. 416).

Les paramètres suivants déterminent le comportement de dimensionnement automatique pour chaque règle d'une stratégie.

Note

Les paramètres répertoriés ici sont basés sur la AWS Management Console pour Amazon EMR. Des options de configuration avancées supplémentaires sont disponibles lorsque vous utilisez l'AWS CLI ou l'API Amazon EMR. Pour plus d'informations sur les options avancées, consultez ConfigurationStratégieDimensionnementSimple du document Amazon EMR API Reference.

- Nombre minimal et nombre maximal d'instances. La contrainte du Maximum instances (Nombre maximal d'instances) spécifie le nombre maximal d'instances Amazon EC2 qui peut figurer dans le groupe d'instances et s'applique à toutes les règles de dimensionnement (augmentation). De même, la contrainte du Minimum instances (Nombre minimal d'instances) spécifie le nombre minimal d'instances Amazon EC2 et s'applique à toutes les règles de dimensionnement (diminution).
- Le Nom de la règle qui doit être unique dans la stratégie.
- L'ajustement de dimensionnement qui détermine le nombre d'instances EC2 à ajouter (pour les règles de dimensionnement d'augmentation) ou à résilier (pour les règles de dimensionnement de diminution).
- La métrique CloudWatch qui est surveillée pour détecter une éventuelle condition d'alarme.
- Un opérateur de comparaison qui est utilisé pour comparer la métrique CloudWatch à la valeur du seuil et déterminer une condition de déclenchement.
- Une période d'évaluation, par incrément de cinq minutes, pour laquelle la métrique CloudWatch doit être dans une condition de déclenchement avant le lancement de l'activité de dimensionnement.
- Un temps de stabilisation qui détermine la durée qui doit s'écouler entre une activité de dimensionnement déclenchée par une règle et le début de l'activité de dimensionnement suivante, quelle que soit la règle qui la déclenche. Quand un groupe d'instances a terminé une activité de dimensionnement et a atteint son état post-dimensionnement, le temps de stabilisation permet aux métriques CloudWatch de déclencher des activités de dimensionnement ultérieures. Pour plus d'informations, consultez Récupérations de dimensionnement automatique dans le Amazon EC2 Auto Scaling Guide de l'utilisateur.

Utilisation de la AWS Management Console pour configurer le dimensionnement automatique

Lorsque vous créez un cluster, vous configurez une stratégie de dimensionnement pour des groupes d'instances à l'aide des options de configuration de cluster avancées. Vous pouvez également créer ou modifier une stratégie de dimensionnement pour un groupe d'instances en service en modifiant les groupes d'instances dans les paramètres Hardware (Matériel) d'un cluster existant.

1. Si vous créez un cluster, dans le Amazon EMR de la console, sélectionnez Créer un cluster, sélectionnez Aller aux options avancées, choisissez des options pour Étape 1 : Logiciels et étapes, puis allez à Étape 2 : Configuration matérielle.

—ou—

Si vous modifiez un groupe d'instances dans un cluster en cours d'exécution, sélectionnez votre cluster dans la liste, puis développez la section Hardware (Matériel).

2. Dans la section Cluster scaling (Dimensionnement du cluster), sélectionnez Enable cluster scaling (Activer le dimensionnement du cluster). Sélectionnez ensuite Create a custom automatic scaling policy (Créer une stratégie de dimensionnement automatique personnalisée).

   Dans le tableau des stratégies de dimensionnement automatique personnalisées, cliquez sur l'icône représentant un crayon qui apparaît dans la ligne du groupe d'instances que vous souhaitez configurer. L'écran des règles Auto Scaling s'ouvre.
3. Saisissez le nombre d'Maximum instances (Instances maximales) que le groupe d'instances doit contenir après avoir été augmenté, puis entrez le nombre d'Minimum instances (Instances minimales) que le groupe d'instances doit contenir après avoir été diminué.
4. Cliquez sur le crayon pour modifier les paramètres de règle, cliquez sur X pour supprimer une règle de la stratégie et cliquez sur Add rule (Ajouter une règle) pour ajouter des règles supplémentaires.
5. Choisissez les paramètres de règle comme indiqué précédemment dans cette rubrique. Pour obtenir des descriptions des métriques CloudWatch disponibles pour Amazon EMR, consultez Dimensions et métriques Amazon EMR dans le Guide de l'utilisateur Amazon CloudWatch.

Utilisation de la AWS CLI pour configurer le dimensionnement automatique

Vous pouvez utiliser les commandes de l'AWS CLI pour Amazon EMR afin de configurer le dimensionnement automatique lors de la création d'un cluster et d'un groupe d'instances. Vous pouvez utiliser une syntaxe raccourcie, en spécifiant la configuration JSON compatible avec les commandes adéquates, ou vous pouvez indiquer un fichier contenant la configuration JSON. Vous pouvez également appliquer une stratégie de dimensionnement automatique à un groupe d'instances existant et supprimer une stratégie de dimensionnement automatique qui a été précédemment appliquée. En outre, vous pouvez récupérer les détails de configuration d'une stratégie de dimensionnement à partir d'un cluster en cours d'exécution.

> **Important**
>
> Lorsque vous créez un cluster qui dispose d'une stratégie de dimensionnement automatique, vous devez utiliser l' `--auto-scaling-role` *`MyAutoScalingRole`* pour spécifier le IAM rôle pour mise à l'échelle automatique. Le rôle par défaut est *`EMR_AutoScaling_DefaultRole`* et peut être créé à l'aide de la commande `create-default-roles`. Le rôle ne peut être ajouté qu'au moment de la création du cluster ; il ne peut pas être ajouté à un cluster existant.

Pour une description détaillée des paramètres disponibles lors de la configuration d'une stratégie dimensionnement automatique, consultez PutAutoScalingPolicy du Amazon EMR API Reference.

Création d'un cluster avec une stratégie de dimensionnement automatique appliquée à un groupe d'instances

Vous pouvez spécifier une configuration de dimensionnement automatique au sein de l'option `--instance-groups` de la commande `aws emr create-cluster`. L'exemple suivant illustre une commande create-cluster où une stratégie de dimensionnement automatique pour le groupe d'instances principal est fournie en ligne. La commande crée une configuration de dimensionnement correspondant à la stratégie d'augmentation par défaut qui s'affiche lorsque vous créez une stratégie de dimensionnement automatique à l'aide de la AWS Management Console pour Amazon EMR. Pour des raisons de concision, une stratégie de diminution n'apparaît pas. Nous vous déconseillons de créer une règle d'augmentation sans définir une règle de diminution.

```
aws emr create-cluster --release-label emr-5.2.0 --service-role
 EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole
 --auto-scaling-role EMR_AutoScaling_DefaultRole  --instance-groups
 Name=MyMasterIG,InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1
 'Name=MyCoreIG,InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,AutoScalingPolicy={Constr
scale-out,Description=Replicates the default scale-out rule in the
 console.,Action={SimpleScalingPolicyConfiguration={AdjustmentType=CHANGE_IN_CAPACITY,ScalingAdjustment
ElasticMapReduce,Period=300,Statistic=AVERAGE,Threshold=15,Unit=PERCENT,Dimensions=[{Key=JobFlowId,Valu
```

La commande suivante illustre l'utilisation de la ligne de commande pour fournir la définition de la stratégie de dimensionnement automatique dans le cadre d'un fichier de configuration de groupe d'instances nommé *instancegroupconfig.json*.

```
aws emr create-cluster --release-label emr-5.2.0 --service-role EMR_DefaultRole --ec2-
attributes InstanceProfile=EMR_EC2_DefaultRole --instance-groups file:///your/path/to/
instancegroupconfig.json --auto-scaling-role EMR_AutoScaling_DefaultRole
```

Avec le contenu du fichier de configuration comme suit :

```
[
{
  "InstanceCount": 1,
  "Name": "MyMasterIG",
  "InstanceGroupType": "MASTER",
  "InstanceType": "m5.xlarge"
},
{
  "InstanceCount": 2,
  "Name": "MyCoreIG",
  "InstanceGroupType": "CORE",
  "InstanceType": "m5.xlarge",
  "AutoScalingPolicy":
    {
     "Constraints":
      {
       "MinCapacity": 2,
       "MaxCapacity": 10
      },
     "Rules":
     [
      {
       "Name": "Default-scale-out",
       "Description": "Replicates the default scale-out rule in the console for YARN
 memory.",
       "Action":{
        "SimpleScalingPolicyConfiguration":{
          "AdjustmentType": "CHANGE_IN_CAPACITY",
          "ScalingAdjustment": 1,
          "CoolDown": 300
        }
       },
       "Trigger":{
        "CloudWatchAlarmDefinition":{
          "ComparisonOperator": "LESS_THAN",
          "EvaluationPeriods": 1,
          "MetricName": "YARNMemoryAvailablePercentage",
          "Namespace": "AWS/ElasticMapReduce",
```

```
          "Period": 300,
          "Threshold": 15,
          "Statistic": "AVERAGE",
          "Unit": "PERCENT",
          "Dimensions":[
             {
               "Key" : "JobFlowId",
               "Value" : "${emr.clusterId}"
             }
          ]
        }
       }
      }
     ]
   }
}
]
```

Ajout d'un groupe d'instances avec une stratégie de dimensionnement automatique à un cluster

Vous pouvez spécifier une configuration de stratégie de dimensionnement à l'aide de l' `--instance-groups` avec l'option `add-instance-groups` de la même façon que lorsque vous utilisez `create-cluster`. L'exemple suivant utilise une référence à un fichier JSON, *`instancegroupconfig.json`*, avec la configuration du groupe d'instances.

```
aws emr add-instance-groups --cluster-id j-1EKZ3TYEVF1S2 --instance-groups file://your/
path/to/instancegroupconfig.json
```

Application d'une stratégie de dimensionnement automatique à un groupe d'instances existant ou modification d'une stratégie appliquée

Utilisez la commande `aws emr put-auto-scaling-policy` pour appliquer une stratégie de dimensionnement automatique à un groupe d'instances existant. Le groupe d'instances doit faire partie d'un cluster qui utilise le rôle IAM de dimensionnement automatique. L'exemple suivant utilise une référence à un fichier JSON, *`autoscaleconfig.json`*, qui spécifie la configuration de stratégie de dimensionnement automatique.

```
aws emr put-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07 --auto-scaling-policy file://your/path/to/autoscaleconfig.json
```

Le contenu du fichier `autoscaleconfig.json` qui définit la même règle d'augmentation comme illustré dans l'exemple précédent, est présenté ci-dessous.

```
{
  "Constraints": {
   "MinCapacity": 2,
   "MaxCapacity": 10
  },
  "Rules": [{
   "Name": "Default-scale-out",
   "Description": "Replicates the default scale-out rule in the console for YARN memory.",
   "Action": {
    "SimpleScalingPolicyConfiguration": {
     "AdjustmentType": "CHANGE_IN_CAPACITY",
     "ScalingAdjustment": 1,
```

```
     "CoolDown": 300
    }
   },
   "Trigger": {
    "CloudWatchAlarmDefinition": {
     "ComparisonOperator": "LESS_THAN",
     "EvaluationPeriods": 1,
     "MetricName": "YARNMemoryAvailablePercentage",
     "Namespace": "AWS/ElasticMapReduce",
     "Period": 300,
     "Threshold": 15,
     "Statistic": "AVERAGE",
     "Unit": "PERCENT",
     "Dimensions": [{
      "Key": "JobFlowId",
      "Value": "${emr.clusterId}"
     }]
    }
   }
  }]
 }
}]
```

Suppression d'une stratégie de dimensionnement automatique d'un groupe d'instances

```
aws emr remove-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07
```

Récupération d'une configuration de stratégie de dimensionnement automatique

La commande `describe-cluster` récupère la configuration de la stratégie dans le bloc InstanceGroup. Par exemple, la commande suivante extrait la configuration pour le cluster avec un ID de cluster de `j-1CWOHP4PI30VJ`.

```
aws emr describe-cluster --cluster-id j-1CWOHP4PI30VJ
```

La commande produit l'exemple de résultat suivant.

```
{
    "Cluster": {
        "Configurations": [],
        "Id": "j-1CWOHP4PI30VJ",
        "NormalizedInstanceHours": 48,
        "Name": "Auto Scaling Cluster",
        "ReleaseLabel": "emr-5.2.0",
        "ServiceRole": "EMR_DefaultRole",
        "AutoTerminate": false,
        "TerminationProtected": true,
        "MasterPublicDnsName": "ec2-54-167-31-38.compute-1.amazonaws.com",
        "LogUri": "s3n://aws-logs-232939870606-us-east-1/elasticmapreduce/",
        "Ec2InstanceAttributes": {
            "Ec2KeyName": "performance",
            "AdditionalMasterSecurityGroups": [],
            "AdditionalSlaveSecurityGroups": [],
            "EmrManagedSlaveSecurityGroup": "sg-09fc9362",
            "Ec2AvailabilityZone": "us-east-1d",
```

```
            "EmrManagedMasterSecurityGroup": "sg-0bfc9360",
            "IamInstanceProfile": "EMR_EC2_DefaultRole"
        },
        "Applications": [
            {
                "Name": "Hadoop",
                "Version": "2.7.3"
            }
        ],
        "InstanceGroups": [
            {
                "AutoScalingPolicy": {
                    "Status": {
                        "State": "ATTACHED",
                        "StateChangeReason": {
                            "Message": ""
                        }
                    },
                    "Constraints": {
                        "MaxCapacity": 10,
                        "MinCapacity": 2
                    },
                    "Rules": [
                        {
                            "Name": "Default-scale-out",
                            "Trigger": {
                                "CloudWatchAlarmDefinition": {
                                    "MetricName": "YARNMemoryAvailablePercentage",
                                    "Unit": "PERCENT",
                                    "Namespace": "AWS/ElasticMapReduce",
                                    "Threshold": 15,
                                    "Dimensions": [
                                        {
                                            "Key": "JobFlowId",
                                            "Value": "j-1CWOHP4PI30VJ"
                                        }
                                    ],
                                    "EvaluationPeriods": 1,
                                    "Period": 300,
                                    "ComparisonOperator": "LESS_THAN",
                                    "Statistic": "AVERAGE"
                                }
                            },
                            "Description": "",
                            "Action": {
                                "SimpleScalingPolicyConfiguration": {
                                    "CoolDown": 300,
                                    "AdjustmentType": "CHANGE_IN_CAPACITY",
                                    "ScalingAdjustment": 1
                                }
                            }
                        },
                        {
                            "Name": "Default-scale-in",
                            "Trigger": {
                                "CloudWatchAlarmDefinition": {
                                    "MetricName": "YARNMemoryAvailablePercentage",
                                    "Unit": "PERCENT",
                                    "Namespace": "AWS/ElasticMapReduce",
                                    "Threshold": 75,
                                    "Dimensions": [
                                        {
                                            "Key": "JobFlowId",
                                            "Value": "j-1CWOHP4PI30VJ"
                                        }
                                    ],
```

```
                                "EvaluationPeriods": 1,
                                "Period": 300,
                                "ComparisonOperator": "GREATER_THAN",
                                "Statistic": "AVERAGE"
                            }
                        },
                        "Description": "",
                        "Action": {
                            "SimpleScalingPolicyConfiguration": {
                                "CoolDown": 300,
                                "AdjustmentType": "CHANGE_IN_CAPACITY",
                                "ScalingAdjustment": -1
                            }
                        }
                    }
                ]
            },
            "Configurations": [],
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Core - 2",
            "ShrinkPolicy": {},
            "Status": {
                "Timeline": {
                    "CreationDateTime": 1479413437.342,
                    "ReadyDateTime": 1479413864.615
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "RunningInstanceCount": 2,
            "Id": "ig-3M16XBE8C3PH1",
            "InstanceGroupType": "CORE",
            "RequestedInstanceCount": 2,
            "EbsBlockDevices": []
        },
        {
            "Configurations": [],
            "Id": "ig-OP62I28NSE8M",
            "InstanceGroupType": "MASTER",
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Master - 1",
            "ShrinkPolicy": {},
            "EbsBlockDevices": [],
            "RequestedInstanceCount": 1,
            "Status": {
                "Timeline": {
                    "CreationDateTime": 1479413437.342,
                    "ReadyDateTime": 1479413752.088
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "RunningInstanceCount": 1
        }
    ],
    "AutoScalingRole": "EMR_AutoScaling_DefaultRole",
    "Tags": [],
    "BootstrapActions": [],
    "Status": {
        "Timeline": {
```

```
                "CreationDateTime": 1479413437.339,
                "ReadyDateTime": 1479413863.666
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Cluster ready after last step completed."
            }
        }
    }
}
```

Redimensionnement manuel d'un cluster en cours d'exécution

Vous pouvez ajouter et supprimer des instances dans les groupes d'instances principaux et de tâches, et des parcs d'instances dans un cluster en cours d'exécution à l'aide de la AWS Management Console, de l'AWS CLI ou de l'API Amazon EMR. Si un cluster utilise des groupes d'instances, vous devez modifier explicitement le nombre d'instances. Si le cluster utilise des parcs d'instances, vous pouvez modifier les unités cibles pour les instances à la demande et les instances Spot. Le parc d'instances ajoute et supprime ensuite des instances afin de répondre à la nouvelle cible. Pour plus d'informations, consultez la section Options de parc d'instances (p. 150). Des applications peuvent utiliser les instances Amazon EC2 nouvellement allouées pour héberger des nœuds dès qu'elles deviennent disponibles. Lorsque les instances sont supprimées, Amazon EMR se termine de manière à ne pas interrompre les tâches et à éviter la perte de données. Pour plus d'informations, consultez la section Mise hors service lors de l'achèvement de la tâche (p. 416).

Redimensionnement d'un cluster à l'aide de la console

Vous pouvez utiliser la console Amazon EMR pour redimensionner un cluster en cours d'exécution.

Pour modifier le nombre d’instances d’un cluster en cours d’exécution à l’aide de la console

1. Dans la page Liste de clusters, choisissez un cluster à redimensionner.
2. Dans la page Détails du cluster, choisissez Matériel.
3. Si votre cluster utilise des groupes d'instances, choisissez Redimensionner dans la colonne Nombre d'instances du groupe d'instances que vous souhaitez redimensionner, saisissez un nouveau nombre, puis sélectionnez la coche verte.

–OU–

Si votre cluster utilise des parcs d'instances, choisissez Redimensionner dans le Capacité allouée colonne, saisissez de nouvelles valeurs pour Unités à la demande et Unités Spot, puis choisissez Redimensionner.

Lorsque vous modifiez le nombre de nœuds, le Statut du groupe d'instances est mis à jour. Lorsque la modification est terminée, le Statut est En cours d'exécution.

Redimensionnement d'un cluster à l'aide de l'AWS CLI

Vous pouvez utiliser l'AWS CLI pour redimensionner un cluster en cours d'exécution. Vous pouvez augmenter ou diminuer le nombre de nœuds de tâches, et vous pouvez augmenter le nombre de nœuds principaux dans un cluster en cours d'exécution. Il est également possible de terminate une instance dans le groupe d'instances principal à l'aide de l' AWS CLI ou l'API. Cela doit être effectué avec précautions. Résiliation une instance du groupe d'instances principal risque de perdre des données et l'instance n'est pas automatiquement remplacée.

Outre le redimensionnement des groupes principaux et de tâches, vous pouvez ajouter un ou plusieurs groupes d'instances de tâches à un cluster en cours d'exécution à l'aide de l'AWS CLI.

Pour redimensionner un cluster en modifiant le nombre d'instances à l'aide de l'AWS CLI

Vous pouvez ajouter des instances au groupe principal ou au groupe de tâches, et vous pouvez supprimer des instances du groupe de tâches à l'aide de la sous-commande `modify-instance-groups` de l'AWS CLI avec le paramètre `InstanceCount`. Pour ajouter des instances au noyau ou aux groupes de tâches, augmentez le `InstanceCount`. Pour réduire le nombre d'instances dans le groupe de tâches, réduisez le `InstanceCount`. La modification du nombre d'instances du groupe de tâches sur 0 supprime toutes les instances, mais pas le groupe d'instances.

- Pour augmenter le nombre d'instances dans le groupe d'instances de tâche de 3 à 4, tapez la commande suivante et remplacez *`ig-31JXXXXXXBTO`* avec l'ID du groupe d'instances.

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-31JXXXXXXBTO,InstanceCount=4
```

Pour récupérer le `InstanceGroupId`, utilisez la sous-commande `describe-cluster`. Le résultat est un objet JSON appelé `Cluster` qui contient l'ID de chaque groupe d'instances. Pour utiliser cette commande, vous avez besoin de l'ID du cluster (que vous pouvez récupérer à l'aide de la commande `aws emr list-clusters` ou de la console). Pour récupérer l'ID du groupe d'instances, tapez la commande suivante et remplacez *`j-2AXXXXXXGAPLF`* avec l'ID de cluster.

```
aws emr describe-cluster --cluster-id j-2AXXXXXXGAPLF
```

Dans l'AWS CLI, vous pouvez également arrêter une instance dans le groupe d'instances principal à l'aide de la sous-commande `--modify-instance-groups`.

> **Warning**
>
> La spécification de `EC2InstanceIdsToTerminate` doit s'effectuer avec précaution. Les instances sont résiliées immédiatement, quel que soit le statut des applications qui s'y exécutent, et l'instance n'est pas automatiquement remplacée. Cela est vrai quelle que soit la configuration Comportement de dimensionnement à la baisse du cluster. Une telle résiliation d'une instance peut entraîner une perte des données et un comportement du cluster imprévisible.

Pour résilier une instance spécifique, vous avez besoin de l'ID du groupe d'instances (retourné par l' `aws emr describe-cluster --cluster-id` sous-commande) et l'ID d'instance (retourné par le `aws emr list-instances --cluster-id` sous-commande), tapez la commande suivante, remplacez *`ig-6RXXXXXX07SA`* avec l'ID du groupe d'instances et remplacez *`i-f9XXXXf2`* avec l'ID d'instance.

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-6RXXXXXX07SA,EC2InstanceIdsToTerminate=i-f9XXXXf2
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Pour redimensionner un cluster en ajoutant des groupes d'instances de tâches à l'aide de l'AWS CLI

Dans l'AWS CLI, vous pouvez ajouter entre 1 et 48 groupes d'instances de tâches à un cluster avec la sous-commande `--add-instance-groups`. Les groupes d'instances de tâches peuvent uniquement être ajoutés à un cluster contenant un groupe d'instances maître et un groupe d'instances principal. Lorsque vous utilisez l'AWS CLI, vous pouvez ajouter jusqu'à cinq groupes d'instances de tâches chaque fois que vous utilisez la sous-commande `--add-instance-groups`.

1. Pour ajouter un groupe d'instances de tâche unique à un cluster, tapez la commande suivante et remplacez *`j-JXBXXXXXX37R`* avec l'ID de cluster.

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-groups
 InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
```

2. Pour ajouter plusieurs groupes d'instances de tâches à un cluster, tapez la commande suivante et remplacez *`j-JXBXXXXXX37R`* avec l'ID de cluster. Vous pouvez ajouter jusqu'à cinq groupes d'instances de tâches dans une seule commande.

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-
groups InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
 InstanceCount=10,InstanceGroupType=task,InstanceType=m5.xlarge
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Interruption d'un redimensionnement

Avec Amazon EMR version 4.1.0 ou ultérieure, vous pouvez émettre un redimensionnement au milieu d'une opération de redimensionnement existante. En outre, vous pouvez arrêter une demande de redimensionnement déjà soumise ou soumettre une nouvelle demande pour remplacer une demande précédente sans attendre qu'elle se termine. Vous pouvez également arrêter un redimensionnement existant à partir de la console ou à l'aide de l'appel d'API `ModifyInstanceGroups` avec le nombre actuel comme nombre cible du cluster.

La capture d'écran suivante illustre un groupe d'instances de tâches qui est en cours de redimensionnement, mais qui peut être arrêté en choisissant Arrêter.

Pour interrompre un redimensionnement à l'aide de l'AWS CLI

Vous pouvez utiliser l'AWS CLI pour arrêter un redimensionnement à l'aide de la sous-commande `modify-instance-groups`. Supposez vous avez six instances dans votre groupe d'instances et que vous voulez passer à 10 instances. Vous décidez plus tard d'annuler cette demande :

- Demande initiale :

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-myInstanceGroupId,InstanceCount=10
```

Seconde demande visant à arrêter la première demande :

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-myInstanceGroupId,InstanceCount=6
```

Note

Comme ce processus est asynchrone, vous pouvez voir les nombres d'instances changer en fonction des demandes d'API précédentes avant que les demandes suivantes soient honorées. Dans le cas d'une réduction, si vous avez du travail en cours d'exécution sur les nœuds, il est possible que le groupe d'instances ne soit pas réduit tant que les nœuds n'ont pas terminé leur travail.

État suspendu

Un groupe d'instances passe à l'état suspendu s'il rencontre trop d'erreurs lors de la tentative de démarrage des nouveaux nœuds de cluster. Par exemple, si de nouveaux nœuds échouent lors de l'exécution d'actions d'amorçage, le groupe d'instances entre dans un SUSPENDU au lieu d'allouer continuellement de nouveaux nœuds. Une fois que vous avez résolu le problème sous-jacent, réinitialisez le nombre voulu de nœuds sur le groupe d'instances du cluster, puis le groupe d'instances reprend l'allocation des nœuds. La modification d'un groupe d'instances indique à Amazon EMR d'essayer de mettre de nouveau en service des nœuds. Aucun nœud en cours d'exécution n'est redémarré ou arrêté.

Dans l'AWS CLI, la sous-commande `list-instances` retourne toutes les instances et leurs états, comme la sous-commande `describe-cluster`. Si Amazon EMR détecte un défaut avec un groupe d'instances, il change l'état du groupe en spécifiant `SUSPENDED`.

Pour réinitialiser un cluster à l'état SUSPENDED à l'aide de l' AWS CLI

Tapez la sous-commande `describe-cluster` avec le paramètre `--cluster-id` afin d'afficher l'état des instances figurant dans votre cluster.

- Pour afficher des informations sur toutes les instances et groupes d'instances d'un cluster, tapez la commande suivante et remplacez *`j-3KVXXXXXXY7UG`* avec l'ID de cluster.

```
aws emr describe-cluster --cluster-id j-3KVXXXXXXY7UG
```

Les données de sortie affichent des informations sur vos groupes d'instances et l'état de ces instances :

```
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1413187781.245,
                "CreationDateTime": 1413187405.356
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting after step completed"
            }
        },
        "Ec2InstanceAttributes": {
            "Ec2AvailabilityZone": "us-west-2b"
        },
        "Name": "Development Cluster",
        "Tags": [],
        "TerminationProtected": false,
        "RunningAmiVersion": "3.2.1",
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187775.749,
                        "CreationDateTime": 1413187405.357
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "MASTER",
                "InstanceGroupType": "MASTER",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3ETXXXXXXFYV8",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187781.301,
                        "CreationDateTime": 1413187405.357
```

```
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "CORE",
                "InstanceGroupType": "CORE",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3SUXXXXXXQ9ZM",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            }
...
}
```

Pour afficher les informations concernant un groupe d'instances particulier, tapez la sous-commande `list-instances` avec les paramètres `--cluster-id` et `--instance-group-types`. Vous pouvez consulter des informations pour les groupes MAÎTRE, PRINCIPAUX et DE TÂCHES.

```
aws emr list-instances --cluster-id j-3KVXXXXXXY7UG --instance-group-types "CORE"
```

Utilisez la sous-commande `modify-instance-groups` avec le paramètre `--instance-groups` pour réinitialiser un cluster dans l'état `SUSPENDED`. L'ID du groupe d'instances est renvoyé par la sous-commande `describe-cluster`.

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-3SUXXXXXXQ9ZM,InstanceCount=3
```

Diminution de capacité des clusters

Les versions 5.1.0 et ultérieures d'Amazon EMR comportent deux options de comportement de diminution de capacité : la résiliation à l'échéance horaire de l'instance pour la facturation Amazon EC2 ou la résiliation à l'achèvement de la tâche. À compter de la version 5.10.0 d'Amazon EMR, le paramètre de résiliation à l'échéance horaire de l'instance est obsolète en raison de l'introduction de la facturation à la seconde dans Amazon EC2. Nous déconseillons de spécifier la résiliation à l'échéance horaire de l'instance dans les versions où cette option est disponible.

> Warning
>
> Si vous utilisez l'AWS CLI pour émettre un `modify-instance-groups` avec `EC2InstanceIdsToTerminate`, ces instances sont immédiatement résiliées, sans considération de ces paramètres et quel que soit le statut des applications qui s'y exécutent. Une telle résiliation d'une instance peut entraîner une perte des données et un comportement du cluster imprévisible.

Lorsque l'option de résiliation à l'achèvement de la tâche est spécifiée, Amazon EMR met en liste noire et vide les tâches provenant des nœuds avant de résilier les instances Amazon EC2. Lorsque l'une ou l'autre des options est spécifiée, Amazon EMR ne résilie pas les instances Amazon EC2 dans les groupes d'instances principaux si cela peut endommager HDFS.

Mise hors service lors de l'achèvement de la tâche

Amazon EMR vous permet de dimensionner votre cluster à la baisse sans affecter votre charge de travail. Amazon EMR déclasse gracieusement YARN, HDFS et d'autres démons sur les nœuds principaux et de tâches lors d'une opération de dimensionnement à la baisse, sans perte de données et sans interruption de tâches. Amazon EMR ne fait que réduire la taille de groupes d'instances si le travail attribué à ces groupes est terminé et qu'ils sont inactifs. Pour la mise hors service de YARN NodeManager, vous pouvez ajuster manuellement le temps pendant lequel un nœud doit patienter avant d'être mis hors service.

Cette durée est définie à l'aide d'une propriété dans la classification de configuration `yarn-site`. Avec les versions 5.12.0 et ultérieures d'Amazon EMR, spécifiez la propriété `yarn.resourcemanager.nodemanager-graceful-decommission-timeout-secs`. Avec les versions antérieures d'Amazon EMR, spécifiez la propriété `yarn.resourcemanager.decommissioning.timeout`.

Si des conteneurs ou des applications YARN s'exécutent encore à l'expiration du délai de mise hors service, la mise hors service du nœud est forcée et YARN replanifie les conteneurs affectés sur d'autres nœuds. La valeur par défaut est de 3 600 secondes (une heure). Vous pouvez affecter une valeur élevée arbitraire à ce délai pour allonger le temps d'attente d'une réduction appropriée. Pour plus d'informations, consultez la section relative au Déclassement gracieux des nœuds YARN dans la documentation Apache Hadoop.

Groupes de nœuds de tâches

Amazon EMR sélectionne intelligemment les instances qui n'exécutent pas de tâches liées à quelconque étape ou application, et les suppriment en premier d'un cluster. Si toutes les instances dans le cluster sont utilisées, Amazon EMR attend que les tâches se terminent sur une instance donnée avant de la supprimer du cluster. Le temps d'attente par défaut est d'1 heure et cette valeur peut être modifiée en définissant `yarn.resourcemanager.decommissioning.timeout`. Amazon EMR utilise le nouveau paramètre de façon dynamique. Vous pouvez lui affecter un nombre élevé arbitraire pour garantir qu'aucune tâche ne se termine au cours de la réduction du cluster.

Groupes de nœuds principaux

Sur les nœuds principaux, les démons YARN NodeManager et HDFS DataNode doivent être mis hors service pour que le groupe d'instances puisse être réduit. Pour YARN, une réduction appropriée garantit qu'un nœud marqué pour mise hors service passe seulement à l'état `DECOMMISSIONED` s'il n'y a aucun

conteneur ni application en attente ou incomplet. La mise hors service se termine immédiatement s'il n'y a aucun conteneur en cours d'exécution sur le nœud au début de la mise hors service.

Pour HDFS, une réduction appropriée garantit que la capacité cible de HDFS est suffisamment grande pour prendre en charge tous les blocs existants. Si la capacité cible n'est pas suffisamment grande, seule une quantité partielle d'instances principales sont mises hors service, de sorte que les nœuds restants pourront gérer les données actuelles résidant dans HDFS. Vous devriez garantir une capacité HDFS supplémentaire pour permettre de continuer la mise hors service. Vous devriez également essayer de réduire au maximum les E/S d'écriture avant d'essayer de réduire les groupes d'instances, car cela peut retarder la fin de l'opération de redimensionnement.

Une autre limite est le facteur de réplication par défaut, `dfs.replication` à l'intérieur `/etc/hadoop/conf/hdfs-site`. Amazon EMR configure la valeur en fonction du nombre d'instances dans le cluster : `1` avec 1 à 3 instances, `2` pour les clusters avec 4 à 9 instances, et `3` pour les clusters avec plus de 10 instances. La réduction appropriée ne vous permet pas de réduire les nœuds principaux au dessous du facteur de réplication HDFS. Ceci a pour but d'empêcher HDFS d'être dans l'incapacité de fermer des fichiers en raison d'un nombre insuffisant de réplicas. Pour contourner cette limite, vous devez diminuer le facteur de réplication et redémarrer le démon NameNode.

Configuration du comportement de diminution d'Amazon EMR

> **Note**
>
> Cette fonctionnalité est disponible uniquement pour les versions 5.1.0 et ultérieures d'Amazon EMR.

Vous pouvez utiliser la AWS Management Console, l'AWS CLI ou l'API Amazon EMR pour configurer le comportement de diminution lorsque vous créez un cluster. Configuration de la réduction à l'aide de l' AWS Management Console est effectué dans le Étape 3 : Paramètres généraux du cluster lorsque vous créez un cluster à l'aide de Options avancées.

Lorsque vous créez un cluster à l'aide de l'AWS CLI, utilisez l'option `--scale-down-behavior` pour spécifier `TERMINATE_AT_INSTANCE_HOUR` ou `TERMINATE_AT_TASK_COMPLETION`.

Cloning a Cluster Using the Console

Vous pouvez utiliser la console Amazon EMR pour cloner un cluster. Cette action copie la configuration du cluster original à utiliser en tant que base pour un nouveau cluster à l'aide de la console Amazon EMR.

Pour cloner un cluster à l'aide de la console

1. Sur la page Liste de clusters, cliquez sur un cluster à cloner.
2. En haut de la page Cluster Details (Détails de clusters), cliquez sur Clone (Cloner).

Dans la boîte de dialogue, choisissez Oui pour inclure les étapes du cluster d'origine dans le cluster cloné. Choisissez Non pour cloner la configuration du cluster d'origine sans inclure aucune étape.

Note

Pour les clusters créés à l'aide de la version AMI 3.1.1 ou d'une version ultérieure (Hadoop 2.x) ou de la version AMI 2.4.8 ou d'une version ultérieure (Hadoop 1.x), si vous clonez un cluster et incluez les étapes, toutes les étapes du système (par exemple, la configuration de Hive) sont clonées, ainsi que les étapes soumises par l'utilisateur, jusqu'à la limite de 1 000 étapes au total. Les étapes plus anciennes qui n'apparaissent plus dans l'historique des étapes de la console ne peuvent pas être clonées. Pour les AMI plus anciennes, seulement 256 étapes peuvent être clonées (y compris les étapes du système). Pour plus d'informations, consultez Submit Work to a Cluster (p. 418),

3. La page Create Cluster (Créer un cluster) affiche une copie de la configuration du cluster d'origine. Passez en revue la configuration, effectuez les modifications nécessaires, puis cliquez sur Create Cluster (Créer un cluster).

Submit Work to a Cluster

Cette section décrit les méthodes permettant de soumettre du travail à un cluster Amazon EMR. Vous pouvez soumettre du travail à un cluster en ajoutant des étapes ou en soumettant de façon interactive des travaux Hadoop au nœud maître. Le nombre maximum autorisé d'étapes en attente (PENDING) et actives (ACTIVE) pour un cluster s'élève à 256. Vous pouvez soumettre des travaux de façon interactive au nœud maître même si 256 étapes actives sont en cours d'exécution sur le cluster. Vous pouvez soumettre un nombre illimité d'étapes pendant la durée de vie d'un cluster de longue durée, mais seules 256 étapes peuvent présenter l'état actif (ACTIVE) ou en attente (PENDING) à un moment donné.

Rubriques

- Work with Steps Using the AWS CLI and Console (p. 418)
- Submit Hadoop Jobs Interactively (p. 423)
- Add More than 256 Steps to a Cluster (p. 424)

Work with Steps Using the AWS CLI and Console

Vous peut ajouter des étapes à un cluster en utilisant AWS Management Console, le AWS CLI, ou Amazon EMR API. Le nombre maximum autorisé d'étapes en attente (PENDING) et actives (ACTIVE) pour un cluster s'élève à 256, ce qui inclut des étapes système telles que l'installation d'Apache Pig, Hive et HBase et la configuration du débogage. Vous pouvez soumettre un nombre illimité d'étapes pendant la durée de vie d'un cluster de longue durée, mais seules 256 étapes peuvent présenter l'état actif (ACTIVE) ou en attente (PENDING) à un moment donné.

Avec Amazon EMR 4.8.0 et versions ultérieures, à l'exception de la version 5.0.0, vous pouvez annuler les étapes qui sont en attente à l'aide d'AWS Management Console, de l'AWS CLI ou de l'API Amazon EMR.

Avec Amazon EMR versions 5.28.0 et ultérieures, vous pouvez annuler les étapes en attente et en cours d'exécution. Vous pouvez également choisir d'exécuter plusieurs étapes en parallèle pour améliorer l'utilisation du cluster et faire des économies.

Rubriques

- Adding Steps to a Cluster Using the Console (p. 419)
- Adding Steps to a Cluster Using the AWS CLI (p. 420)
- Considerations for Running Multiple Steps in Parallel (p. 421)
- Viewing Steps (p. 422)

- Canceling Steps (p. 422)

Adding Steps to a Cluster Using the Console

Vous pouvez ajouter des étapes à un cluster à l'aide de l'AWS CLI, du kit SDK Amazon EMR ou de la AWS Management Console.

Pour ajouter des étapes lors de la création du cluster

AWS Management Console vous permet d'ajouter des étapes à un cluster lors de sa création.

1. Dans la console Amazon EMR, choisissez Create Cluster - Advanced Options (Créer un cluster - Options avancées).
2. Sur le Étape 1 : Logiciels et étapes page, pour Étapes (facultatif), sélectionnez Exécutez plusieurs étapes en parallèle pour améliorer l'utilisation du cluster et réduire les coûts. La valeur par défaut du niveau de simultanéité est 10. Vous pouvez choisir entre 2 et 256 étapes pouvant s'exécuter en parallèle.

   > **Note**
   >
   > L'exécution de plusieurs étapes en parallèle n'est prise en charge qu'avec Amazon EMR version 5.28.0 ou ultérieure.
3. Pour After last step completes (Après la fin de la dernière étape), choisissez Cluster enters waiting state (Le cluster passe à l'état d'attente) ou Auto-terminate the cluster (Résilier automatiquement le cluster).
4. Choisissez Step type (Type d'étape), puis Add step (Ajouter une étape).
5. Entrez les valeurs appropriées dans les champs de la boîte de dialogue Add step (Ajouter une étape). Les options diffèrent selon le type d'étape. Si vous avez activé Run multiple steps in parallel to improve cluster utilization and save cost (Exécuter plusieurs étapes en parallèle pour améliorer l'utilisation du cluster et réduire les coûts), la seule option disponible pour Action on failure (Action en cas d'échec) est Continue (Continuer). Ensuite, choisissez Add (Ajouter).

Pour ajouter des étapes à un cluster en cours d'exécution

À l'aide de l'AWS Management Console, vous pouvez ajouter des étapes à un cluster de longue durée, c'est-à-dire un cluster avec l'option d'arrêt automatique (auto-terminate) désactivée.

1. Dans la console Amazon EMR, dans la page Cluster List (Liste de clusters), sélectionnez le lien correspondant à votre cluster.
2. Sur la page Cluster Details (Détails du cluster), choisissez l'onglet Steps (Étapes).
3. Sous l'onglet Steps (Étapes) choisissez Add step (Ajouter une étape).
4. Entrez les valeurs appropriées dans les champs de la boîte de dialogue Add step (Ajouter une étape), puis cliquez sur Add (Ajouter). Les options varient selon le type d'étape.

Pour modifier le niveau de simultanéité des étapes dans un cluster en cours d'exécution

À l'aide de l'AWS Management Console, vous pouvez modifier le niveau de simultanéité des étapes dans un cluster en cours d'exécution.

> **Note**
>
> L'exécution de plusieurs étapes en parallèle n'est prise en charge qu'avec Amazon EMR version 5.28.0 ou ultérieure.

1. Dans la console Amazon EMR, dans la page Cluster List (Liste de clusters), sélectionnez le lien correspondant à votre cluster.

2. Sur la page Cluster Details (Détails du cluster), choisissez l'onglet Steps (Étapes).
3. Pour Concurrency (Simultanéité), choisissez Change (Modifier). Sélectionnez une nouvelle valeur pour le niveau de simultanéité des étapes, puis enregistrez.

Adding Steps to a Cluster Using the AWS CLI

Les procédures suivantes illustrent l'ajout d'étapes à un cluster nouvellement créé et à un cluster en cours d'exécution à l'aide de l'AWS CLI. Dans les deux exemples, la sous-commande `--steps` est utilisée pour ajouter des étapes au cluster.

Pour ajouter des étapes lors de la création du cluster

- Tapez la commande suivante pour créer un cluster et ajouter une étape Apache Pig. Remplacez *`myKey`* avec le nom de votre Amazon EC2 paire de clés et remplacer *`mybucket`* avec le nom de votre Amazon S3 seau.

  - Linux, UNIX, and macOS

```
aws emr create-cluster --name "Test cluster" --ami-version 2.4 --applications
 Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--steps Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/
scripts/pigscript.pig,-p,INPUT=s3://mybucket/inputdata/,-p,OUTPUT=s3://mybucket/
outputdata/,$INPUT=s3://mybucket/inputdata/,$OUTPUT=s3://mybucket/outputdata/]
```

  - Windows

```
aws emr create-cluster --name "Test cluster" --ami-version 2.4 --applications
 Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey --
instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge --steps
 Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/
scripts/pigscript.pig,-p,INPUT=s3://mybucket/inputdata/,-p,OUTPUT=s3://mybucket/
outputdata/,$INPUT=s3://mybucket/inputdata/,$OUTPUT=s3://mybucket/outputdata/]
```

  > **Note**
  >
  > La liste des arguments change en fonction du type d'étape.

  Par défaut, le niveau de codevise de l'étape est `1`. Vous pouvez définir le niveau de codevise d'étape en utilisant le `StepConcurrencyLevel` lorsque vous créez un cluster.

  Le résultat est un identifiant de cluster similaire au suivant.

```
{
    "ClusterId": "j-2AXXXXXXGAPLF"
}
```

Pour ajouter une étape à un cluster en cours d'exécution

- Tapez la commande suivante pour ajouter une étape à un cluster en cours d'exécution. Remplacez *`j-2AXXXXXXGAPLF`* avec votre ID de cluster et remplacez *`mybucket`* avec votre Amazon S3 nom du seau.

```
aws emr add-steps --cluster-id j-2AXXXXXXGAPLF --steps Type=PIG,Name="Pig
 Program",Args=[-f,s3://mybucket/scripts/pigscript.pig,-p,INPUT=s3://
mybucket/inputdata/,-p,OUTPUT=s3://mybucket/outputdata/,$INPUT=s3://mybucket/
inputdata/,$OUTPUT=s3://mybucket/outputdata/]
```

Le résultat est un identifiant d'étape similaire au suivant.

```
{
    "StepIds": [
 "s-Y9XXXXXXAPMD"
    ]
}
```

Pour modifier le paramètre StepConcurrencyLevel dans un cluster en cours d'exécution

1. Dans un cluster en cours d'exécution, vous pouvez modifier le paramètre StepConcurrencyLevel à l'aide de l'API `ModifyCluster`. Par exemple, tapez la commande suivante pour augmenter l'échelle de nivellementà `10`. Remplacer *`j-2AXXXXXXGAPLF`* avec votre ID de cluster.

```
aws emr modify-cluster --cluster-id j-2AXXXXXXGAPLF --step-concurrency-level 10
```

2. La sortie est similaire à ce qui suit.

```
{
"StepConcurrencyLevel": 10
}
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Considerations for Running Multiple Steps in Parallel

- When you select a step concurrency level for your cluster, you must consider whether or not the master node instance type meets the memory requirements of user workloads. The main step executer process runs on the master node for each step. Running multiple steps in parallel requires more memory and CPU utilization from the master node than running one step at a time.
- To achieve complex scheduling and resource management of concurrent steps, you can use YARN scheduling features such as `FairScheduler` or `CapacityScheduler`. For example, you can use `FairScheduler` with a `queueMaxAppsDefault` set to prevent more than a certain number of jobs from running at a time.
- The step concurrency level is subject to the configurations of resource managers. For example, if YARN is configured with only a parallelism of `5`, then you can only have five YARN applications running in parallel even if the `StepConcurrencyLevel` is set to `10`. For more information about configuring resource managers, see Configuring Applications in the Amazon EMR Guide de version.
- You can use EMR automatic scaling to scale up and down based on the YARN resources to prevent resource contention. For more information, see Using Automatic Scaling in Amazon EMR in the Amazon EMR Guide de gestion.
- When you decrease the step concurrent level, EMR allows any running steps to complete before reducing the number of steps. If the resources are exhausted because the cluster is running too many concurrent steps, we recommend manually canceling any running steps to free up resources.

Viewing Steps

Le nombre total d'enregistrements d'étapes que vous pouvez afficher (quel que soit le statut) est de 1 000. Ce total comprend les étapes système et d'utilisateur. Lorsque le statut des étapes d'utilisateur devient TERMINÉ ou ÉCHEC, des étapes d'utilisateur supplémentaires peuvent être ajoutées au cluster, dans la limite de 1 000 étapes. Une fois que 1 000 étapes ont été ajoutées à un cluster, la soumission d'étapes supplémentaires entraîne la suppression d'enregistrements d'étapes d'utilisateur plus anciens. Ces enregistrements ne sont pas supprimés des fichiers journaux. Ils sont supprimés de l'affichage de la console et ils n'apparaissent pas lorsque vous utilisez l'AWS CLI ou l'API pour récupérer les informations de cluster. Les enregistrements d'étapes système ne sont jamais supprimés.

Les informations d'étape que vous pouvez visualiser dépendent du mécanisme utilisé pour récupérer les informations de cluster. Le tableau suivant indique les informations d'étape renvoyées par chaque option disponible.

Option	DescribeJobFlow ou --describe --jobflow	ListSteps ou list-steps
Kit SDK	256 étapes	1 000 étapes
Interface de ligne de commande Amazon EMR	256 étapes	: ND
AWS CLI	: ND	1 000 étapes
API	256 étapes	1 000 étapes

Canceling Steps

Vous pouvez annuler des étapes en attente et en cours d'exécution à l'aide d'AWS Management Console, de l'AWS CLI ou de l'API Amazon EMR.

Pour annuler des étapes à l'aide de la AWS Management Console

1. Dans la console Amazon EMR, dans la page Cluster List (Liste de clusters), cliquez sur le lien correspondant au cluster.
2. Sur la page Cluster Details (Détails de cluster), développez la section Étapes.
3. Pour chaque étape que vous souhaitez annuler, sélectionnez l'étape dans la liste Steps (Étapes). Choisissez ensuite Cancel step (Annuler l'étape).
4. Dans la boîte de dialogue Cancel step (Annuler l'étape) conservez l'option par défaut Cancel the step and wait for it to exit (Annuler l'étape et attendre qu'elle se termine). Si vous voulez mettre arrêter immédiatement l'étape sans attendre la fin des processus, choisissez Cancel the step and force it to exit (Annuler l'étape et la forcer à se terminer).
5. Choisissez Cancel step (Annuler l'étape).

Pour annuler des étapes à l'aide de la AWS CLI

- Utilisez la commande `aws emr cancel-steps`, en précisant le cluster et les étapes à annuler. L'exemple suivant représente une commande AWS CLI pour annuler en deux étapes.

```
aws emr cancel-steps --cluster-id j-2QUAJ7T3OTEI8 --step-
ids s-3M8DKCZYYN1QE,s-3M8DKCZYYN1QE
```

Avec Amazon EMR version 5.28.0, vous pouvez choisir l'une des deux options d'annulation suivantes pour le paramètre `StepCancellationOption` lors de l'annulation d'étapes.

- `SEND_INTERRUPT_SIGNAL`– This is the default option. When a step cancellation request is received, EMR will send a SIGTERM signal to the step child process until it ends.
- `TERMINATE_PROCESS` – When this option is selected, EMR sends a SIGKILL signal to the step child process.

Submit Hadoop Jobs Interactively

En plus d'ajouter des étapes à un cluster, vous pouvez vous connecter au nœud maître à l'aide d'un client SSH ou de l'AWS CLI, et soumettre de façon interactive des travaux Hadoop. Par exemple, vous pouvez utiliser pour PuTTY établir une connexion SSH avec le nœud principal et soumettre des requêtes Hive interactives. Les requêtes sont compilées en une ou plusieurs tâches Hadoop.

Vous pouvez envoyer des tâches Hadoop de manière interactive en établissant une connexion SSH au nœud principal. Établissez la connexion SSH à l'aide d'un client SSH, tel que PuTTY ou OpenSSH, ou à l'aide de la sous-commande SSH de l'AWS CLI. Vous pouvez soumettre des travaux de façon interactive au nœud maître même si 256 étapes actives sont en cours d'exécution sur le cluster. Notez toutefois que les enregistrements consignés associés à des tâches soumises de façon interactive sont inclus dans la section step created jobs du journal du contrôleur de l'étape en cours d'exécution. Pour plus d'informations sur les journaux d'étape, consultez View Log Files (p. 333).

Les exemples suivants illustrent comment soumettre de façon interactive des travaux Hadoop et des travaux Hive au nœud maître. Le processus de soumission des travaux pour d'autres infrastructures de programmation (telles que Pig) est similaire à ces exemples.

Pour soumettre des travaux Hadoop de façon interactive à l'aide de l'AWS CLI

- Vous pouvez soumettre des travaux Hadoop de façon interactive à l'aide de l'AWS CLI en établissant une connexion SSH dans la commande d'AWS CLI (à l'aide de la sous-commande `ssh`). Pour copier un fichier JAR à partir de votre machine Windows locale dans le système de fichiers du nœud principal, tapez la commande suivante. Remplacez *`j-2A6HXXXXXXL7J`* avec votre ID de cluster, remplacez *`mykey.ppk`* avec le nom de votre fichier de paires clés, et remplacez *`myjar.jar`* avec le nom de votre fichier JAR.

  ```
  aws emr put --cluster-id j-2A6HXXXXXXL7J --key-pair-file "C:\Users\username\Desktop
  \Keys\mykey.ppk" --src "C:\Users\username\myjar.jar"
  ```

  Pour créer une connexion SSH et soumettre le travail Hadoop `myjar.jar`, tapez la commande suivante.

  ```
  aws emr ssh --cluster-id j-2A6HXXXXXXL7J --key-pair-file "C:\Users\username\Desktop
  \Keys\mykey.ppk" --command "hadoop jar myjar.jar"
  ```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Pour soumettre de façon interactive des travaux Hive à l'aide de l'AWS CLI

En plus de soumettre des travaux au nœud principal via des fichiers JAR, vous pouvez soumettre des travaux en interagissant avec l'une des infrastructures de programmation Hadoop en cours d'exécution sur le nœud principal. Par exemple, vous pouvez soumettre de façon interactive des requêtes Hive ou des transformations Pig dans la ligne de commande, ou vous pouvez soumettre des scripts au cluster en vue de leur traitement. Vos commandes ou scripts sont ensuite compilés en une ou plusieurs tâches Hadoop.

La procédure suivante illustre l'exécution d'un script Hive à l'aide de l'AWS CLI.

1. Si Hive n'est pas installé sur le cluster, tapez la commande suivante pour installer. Remplacez *j-2A6HXXXXXXL7J* avec votre ID de cluster.

```
aws emr install-applications --cluster-id j-2A6HXXXXXXL7J --apps Name=Hive
```

2. Créez un fichier de script Hive contenant les requêtes ou les commandes à exécuter. L'exemple de script suivant, nommé `my-hive.q`, crée deux tables, `aTable` et `anotherTable`, et copie le contenu de `aTable` dans `anotherTable` en y remplaçant toutes les données.

```
---- sample Hive script file: my-hive.q ----
create table aTable (aColumn string) ;
create table anotherTable like aTable;
insert overwrite table anotherTable select * from aTable
```

3. Tapez les commandes suivantes pour exécuter le script à partir de la ligne de commande à l'aide de la sous-commande `ssh`.

   Pour copier `my-hive.q` à partir d'une machine Windows vers votre cluster, tapez la commande suivante. Remplacez *j-2A6HXXXXXXL7J* avec votre ID de cluster et remplacez *mykey.ppk* avec le nom de votre fichier de paires clés.

```
aws emr put --cluster-id j-2A6HXXXXXXL7J --key-pair-file "C:\Users\username\Desktop
\Keys\mykey.ppk" --src "C:\Users\username\my-hive.q"
```

   Pour créer une connexion SSH et soumettre le script Hive `my-hive.q`, tapez la commande suivante.

```
aws emr ssh --cluster-id j-2A6HXXXXXXL7J --key-pair-file "C:\Users\username\Desktop
\Keys\mykey.ppk" --command "hive -f my-hive.q"
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans l'AWS CLI, consultez https://docs.aws.amazon.com/cli/latest/reference/emr.

Add More than 256 Steps to a Cluster

Avec l'AMI 3.1.1 (Hadoop 2.x) ou version ultérieure et l'AMI 2.4.8 (Hadoop 1.x) ou version ultérieure, vous pouvez soumettre un nombre illimité d'étapes sur la durée de vie d'un cluster de longue durée. Toutefois, seules 256 étapes peuvent être à l'état PENDING ou ACTIVE à un moment donné. Pour les versions antérieures d'AMI, le nombre total d'étapes qui peuvent être traitées par un cluster est limité à 256, y compris les étapes système telles que l'installation de Hive ou de Pig. Pour plus d'informations, consultez Submit Work to a Cluster (p. 418),

Vous pouvez utiliser l'une des méthodes suivantes pour dépasser la limite de 256 étapes dans les versions d'AMI antérieures à 3.1.1 et 2.4.8 :

1. Configurez chaque étape pour qu'elle soumette plusieurs travaux à Hadoop. Cela ne vous permet pas d'avoir un nombre illimité d'étapes dans les versions d'AMI antérieures à 3.1.1 et 2.4.8, mais c'est la solution la plus simple si vous avez besoin d'un nombre fixe d'étapes supérieur à 256.
2. Ecrivez un programme de flux de travail qui s'exécute dans une étape sur un cluster de longue durée et soumet des travaux à Hadoop. Le programme de flux de travail qui peut exécuter l'une des opérations suivantes :
   - Listen to an Amazon SQS queue to receive information about new steps to run.
   - Check an Amazon S3 bucket on a regular schedule for files containing information about the new steps to run.

3. Écrivez un programme de flux de travail qui s'exécute sur une instance Amazon EC2 en dehors d'Amazon EMR et soumet des travaux à vos clusters de longue durée à l'aide de SSH.
4. Connectez-vous à votre cluster de longue durée via SSH et soumettez des travaux Hadoop à l'aide de l'API Hadoop. Pour plus d'informations, consultez http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/JobClient.html.
5. Connectez-vous au nœud principal et soumettez des tâches au cluster. Vous pouvez vous connecter au nœud principal à l'aide d'un client SSH (tel que PuTTY ou OpenSSH) et soumettre manuellement des tâches au cluster, ou utiliser la sous-commande `ssh` dans l'AWS CLI pour vous connecter et soumettre les tâches. Pour plus d'informations sur l'établissement d'une connexion SSH au nœud maître, consultez Connexion au nœud maître à l'aide de SSH (p. 368). Pour plus d'informations sur la soumission de façon interactive de travaux Hadoop, consultez Submit Hadoop Jobs Interactively (p. 423).

Automate Recurring Clusters with AWS Data Pipeline

AWS Data Pipeline est un service qui automatise le déplacement et la transformation de données. Vous pouvez l'utiliser pour planifier le transfert de données d'entrée dans Amazon S3 et pour planifier le lancement de clusters pour traiter ces données. Imaginons par exemple que vous avez un serveur web qui enregistre des journaux de trafic. Si vous souhaitez exécuter un cluster hebdomadaire pour analyser les données de trafic, vous pouvez utiliser AWS Data Pipeline pour planifier ces clusters. AWS Data Pipeline est un flux de travail piloté par les données, ce qui permet qu'une tâche (lancer le cluster) soit dépendante d'une autre tâche (déplacer les données d'entrée vers Amazon S3). Il possède également une puissante fonctionnalité pour les nouvelles tentatives.

Pour plus d'informations sur AWS Data Pipeline, consultez le AWS Data Pipeline Guide du développeur, en particulier les didacticiels relatifs à Amazon EMR :

- Tutorial: Launch an Amazon EMR Job Flow
- Getting Started: Process Web Logs with AWS Data Pipeline, Amazon EMR, and Hive
- Tutorial: Amazon DynamoDB Import and Export Using AWS Data Pipeline

Troubleshoot a Cluster

Un cluster hébergé par Amazon EMR s'exécute dans un écosystème complexe composé de plusieurs types de logiciels open source, d'un code d'application personnalisé et d'Amazon Web Services. Un problème avec l'un de ces éléments peut entraîner l'échec ou une exécution plus lente que prévue du cluster. Les rubriques suivantes vous aideront à comprendre le problème rencontré par votre cluster et vous apporteront des suggestions sur la résolution du problème.

Rubriques

- What Tools are Available for Troubleshooting? (p. 426)
- Viewing and Restarting Amazon EMR and Application Processes (Daemons) (p. 428)
- Troubleshoot a Failed Cluster (p. 429)
- Résolution des problèmes de rapidité d'un cluster (p. 433)
- Common Errors in Amazon EMR (p. 440)
- Dépannage d'un cluster Lake Formation (version bêta) (p. 456)

Lorsque vous développez une nouvelle application Hadoop, nous vous recommandons d'activer le débogage et de traiter un sous-ensemble réduit, mais représentatif de vos données, pour tester l'application. Vous pouvez également exécuter l'application étape par étape pour tester chaque étape séparément. Pour plus d'informations, consultez Configure Cluster Logging and Debugging (p. 166) et Step 5: Test the Cluster Step by Step (p. 432).

What Tools are Available for Troubleshooting?

Il existe plusieurs outils que vous pouvez utiliser pour obtenir des informations sur votre cluster afin de déterminer ce qui s'est passé. Certains outils doivent être initialisés lorsque vous lancez le cluster. D'autres sont disponibles pour chaque cluster.

Rubriques

- Tools to Display Cluster Details (p. 426)
- Tools to View Log Files (p. 427)
- Tools to Monitor Cluster Performance (p. 427)

Tools to Display Cluster Details

Vous pouvez utiliser la AWS Management Console, l'AWS CLI ou l'API EMR pour récupérer des informations détaillées sur un cluster EMR et sur l'exécution d'un travail. Pour plus d'informations sur l'utilisation de la AWS Management Console et de l'AWS CLI, consultez View Cluster Status and Details (p. 319).

Amazon EMR Console Details Pane

Dans la liste Clusters de la console Amazon EMR, vous pouvez voir des informations de haut niveau sur le statut de chaque cluster de votre compte et de votre région. La liste affiche tous les clusters que vous avez lancés au cours des deux derniers mois, qu'ils soient actifs ou résiliés. Dans la liste Clusters, vous pouvez sélectionner un Nom de cluster pour en visualiser les informations détaillées. Ces informations sont organisées en différentes catégories pour faciliter la navigation.

Les interfaces utilisateur d'application disponible dans la page des détails du cluster peuvent être particulièrement utile pour la résolution des problèmes. Il fournit le statut des applications YARN et pour certaines, comme les applications Spark, vous pouvez explorer les différentes métriques et facettes, telles que les travaux, les phases et les exécuteurs. Pour plus d'informations, consultez Afficher l'historique de l'application (p. 327), Cette fonctionnalité est uniquement disponible dans les versions Amazon EMR 5.8.0 et ultérieures.

Amazon EMR Command Line Interface

Vous pouvez rechercher les détails relatifs à un cluster à partir de l'interface de ligne de commande à l'aide de l'argument `--describe`.

Amazon EMR API

Vous pouvez rechercher les détails relatifs à un cluster à partir de l'API à l'aide de l'action `DescribeJobFlows`.

Tools to View Log Files

Amazon EMR et Hadoop génèrent tous deux des fichiers journaux lorsque le cluster s'exécute. Vous pouvez accéder à ces fichiers journaux grâce à différents outils, en fonction de la configuration que vous avez spécifiée lorsque vous avez lancé le cluster. Pour plus d'informations, consultez Configure Cluster Logging and Debugging (p. 166),

Log Files on the Master Node

Chaque cluster publie des fichiers journaux dans le répertoire /mnt/var/log/ sur le nœud maître. Ces fichiers journaux sont disponibles uniquement pendant l'exécution du cluster.

Log Files Archived to Amazon S3

Si vous lancez le cluster et spécifiez un chemin d'accès au journal Amazon S3, le cluster copie les fichiers journaux stockés dans /mnt/var/log/ sur le nœud principal dans Amazon S3 toutes les 5 minutes. Vous avez ainsi la garantie de pouvoir accéder aux fichiers journaux même après la mise hors service du cluster. Etant donné que les fichiers sont archivés toutes les 5 minutes, les dernières minutes d'un cluster mis hors service soudainement peuvent ne pas être disponibles.

Tools to Monitor Cluster Performance

Amazon EMR propose plusieurs outils pour surveiller les performances de votre cluster.

Hadoop Web Interfaces

Chaque cluster publie un ensemble d'interfaces web sur le nœud maître, qui contient des informations sur le cluster. Vous pouvez accéder à ces pages web à l'aide d'un tunnel SSH pour les connecter sur le nœud maître. Pour plus d'informations, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372),

CloudWatch Metrics

Chaque cluster signale des métriques à CloudWatch. CloudWatch est un service web qui assure le suivi des métriques et qui vous permet de définir des alarmes sur ces métriques. Pour plus d'informations, consultez Surveillance des métriques avec CloudWatch (p. 348),

Viewing and Restarting Amazon EMR and Application Processes (Daemons)

Lorsque vous dépannez un cluster, vous pouvez souhaiter afficher les processus en cours d'exécution. Dans certains cas, vous pouvez aussi trouver utile d'arrêter ou de redémarrer des processus : par exemple, après que vous avez changé une configuration ou que vous avez noté un problème avec un processus donné une fois que vous avez analysé les fichiers journaux et les messages d'erreur.

Il existe deux types de processus qui s'exécutent sur un cluster : les processus Amazon EMR (par exemple, le contrôleur d'instance et le transmetteur de journaux) et les processus associés aux applications installées sur le cluster (par exemple, hadoop-hdfs-namenode et hadoop-yarn-resourcemanager).

Pour utiliser directement les processus sur un cluster, vous vous connectez au nœud maître. Pour plus d'informations, consultez Connexion au cluster (p. 367),

Viewing Running Processes

Si vous utilisez une version d'Amazon EMR 4.x ou ultérieure, les versions de l'application sont emballées à l'aide d'un système basé sur Apache Bigtop et, par conséquent, les processus d'application sont configurés via les scripts .conf sous le système d'initialisation upstart. D'autre part, les processus Amazon EMR sont configurés à l'aide de SysV (scripts init.d), rétrocompatible avec upstart.

To view a list of running Amazon EMR processes

- Tapez la commande suivante (sans le symbole `$`, qui indique l'invite de commande Linux) :

  ```
  $  ls /etc/init.d/
  ```

  La commande retourne une liste des processus Amazon EMR en cours d'exécution similaire à ce qui suit :

  ```
  acpid          cloud-init-local          instance-controller          ntpd
  ```

To view a list of processes associated with application releases

- Saisissez la commande suivante.

  ```
  $  ls /etc/init/
  ```

  La commande retourne une liste des processus d'application en cours d'exécution similaire à ce qui suit :

  ```
  control-alt-delete.conf          hadoop-yarn-resourcemanager.conf                hive-
  metastore.conf
  ```

Restarting Processes

Après que vous avez déterminé les processus en cours d'exécution, vous pouvez les arrêter et les redémarrer si nécessaire. La façon dont vous commencez et arrêtez un service dépend du fait que c'est Amazon EMR service ou service associé à une application.

To restart a process associé à une version d'application

1. Entrez la commande suivante pour arrêter le processus, en remplaçant *processname* par le nom du processus retourné par la commande `ls` dans la procédure ci-dessus :

```
$ sudo /etc/init.d/processname stop
```

   Par exemple: `sudo /etc/init.d/hadoop-hdfs-namenode stop`

2. Tapez la commande suivante pour redémarrer le processus :

```
$ sudo /etc/init.d/processname start
```

   Par exemple : `sudo /etc/init.d/hadoop-hdfs-namenode start`.

To restart an Amazon EMR process

1. Entrez la commande suivante pour arrêter le processus, en remplaçant *processname* par le nom du processus retourné par la commande `ls` dans la procédure ci-dessus :

```
$ sudo /sbin/stop processname
```

   Par exemple : `sudo /sbin/stop instance-controller`.

2. Tapez la commande suivante pour redémarrer le processus :

```
$ sudo sbin/start processname
```

   Par exemple : `sudo sbin/start instance-controller`.

   Note

   Le `sbin/start, stop` et `restart` les commandes sont des liens vers `/sbin/intictl`. Pour plus d'informations sur `initctl`, voir la page initctl man en saisissant `man initctl` à l'invite de commande.

Troubleshoot a Failed Cluster

Cette section vous guide tout au long du processus de dépannage d'un cluster qui a échoué. Cela signifie que le cluster s'est arrêté avec un code d'erreur. Si le cluster est encore en cours d'exécution, mais prend beaucoup de temps pour retourner des résultats, consultez plutôt Résolution des problèmes de rapidité d'un cluster (p. 433).

Rubriques

- Step 1: Gather Data About the Issue (p. 430)
- Step 2: Check the Environment (p. 430)
- Step 3: Look at the Last State Change (p. 431)
- Step 4: Examine the Log Files (p. 431)
- Step 5: Test the Cluster Step by Step (p. 432)

Step 1: Gather Data About the Issue

La première étape de résolution des problèmes d'un cluster consiste à collecter des informations sur l'incident, ainsi que sur l'état et la configuration actuels du cluster. Ces informations seront utilisées dans les étapes suivantes pour confirmer ou infirmer les causes possibles du problème.

Define the Problem

Pour commencer, définissez clairement le problème. Il convient de vous poser quelques questions :

- Quel était le résultat attendu ? Que s'est-il passé à la place ?
- Quand ce problème s'est-il produit pour la première fois ? A quelle fréquence s'est-il produit depuis ?
- Ai-je changé quelque chose dans la configuration ou l'exécution de mon cluster ?

Cluster Details

Les détails de cluster suivants sont utiles pour aider à localiser les problèmes. Pour en savoir plus sur la collecte de ces informations, consultez la page View Cluster Status and Details (p. 319).

- Identificateur du cluster (également appelé identificateur de flux de travail).
- Région et zone de disponibilité où le cluster a été lancé.
- État du cluster, notamment les détails du dernier changement d'état.
- Type et nombre d'instances EC2 spécifiés pour les nœuds maîtres, principaux et de tâches.

Step 2: Check the Environment

Amazon EMR opère dans le cadre d'un écosystème de services web et de logiciels open source. Des éléments affectant ces dépendances peuvent affecter les performances d'Amazon EMR.

Rubriques

- Check for Service Outages (p. 430)
- Check Usage Limits (p. 430)
- Check the Release Version (p. 431)
- Check the Amazon VPC Subnet Configuration (p. 431)

Check for Service Outages

Amazon EMR utilise plusieurs Amazon Web Services en interne. Il exécute des serveurs virtuels sur Amazon EC2, stocke les données et les scripts sur Amazon S3, indexe les fichiers journaux dans Amazon SimpleDB et communique les métriques à CloudWatch. Les événements qui perturbent ces services sont rares mais, lorsqu'ils se produisent, peuvent provoquer des problèmes dans Amazon EMR.

Avant d'aller plus loin, consultez le Tableau de bord de l'état des services. Examinez la région où vous avez lancé votre cluster pour voir s'il existe des événements de perturbation dans l'un de ces services.

Check Usage Limits

Si vous lancez un cluster volumineux et que vous avez lancé de nombreux clusters simultanément, ou si vous êtes un utilisateur IAM partageant un compte AWS avec d'autres utilisateurs, le cluster est peut-être tombé en panne parce que vous avez dépassé une limite de service AWS.

Amazon EC2 limite le nombre d'instances de serveur virtuel en cours d'exécution sur une seule région AWS à 20 instances réservées ou à la demande. Si vous lancez un cluster avec plus de 20 nœuds, ou que vous lancez un cluster qui entraîne le dépassement du nombre total d'instances EC2 actives sur votre compte AWS au-delà de 20, le cluster ne pourra pas lancer toutes les instances EC2 nécessaires et risque de tomber en panne. Dans ce cas, Amazon EMR renvoie une erreur `EC2 QUOTA EXCEEDED`. Vous pouvez demander à ce qu'AWS augmente le nombre d'instances EC2 que vous pouvez exécuter sur votre compte en envoyant une demande d'augmentation de la limite d'instances Amazon EC2.

Une autre raison pouvant provoquer le dépassement de vos limites d'utilisation est le délai entre l'arrêt d'un cluster et la libération de la totalité de ses ressources. En fonction de sa configuration, l'arrêt complet d'un cluster et la libération des ressources allouées peut prendre de 5 à 20 minutes. Si vous obtenez une erreur `EC2 QUOTA EXCEEDED` lorsque vous tentez de lancer un cluster, il est possible que certaines ressources provenant d'un cluster récemment arrêté n'aient pas encore été libérées. Dans ce cas, vous pouvez soit demander à ce que votre quota Amazon EC2 soit augmenté, soit attendre 20 minutes et relancer le cluster.

Amazon S3 limite le nombre de compartiments créés sur un compte à 100. Si votre cluster crée un compartiment qui dépasse cette limite, la création du compartiment échoue et peut provoquer l'arrêt du cluster.

Check the Release Version

Comparez le libellé de la version que vous avez utilisée pour lancer le cluster avec la dernière version d'Amazon EMR. Chaque version d'Amazon EMR inclut des améliorations telles que de nouvelles applications et fonctions, des correctifs et des correctifs de bogues. Le problème qui affecte votre cluster a peut-être déjà été corrigé dans la dernière version. Si possible, réexécutez votre cluster à l'aide de la dernière version.

Check the Amazon VPC Subnet Configuration

Si votre cluster a été lancé dans un sous-réseau Amazon VPC, le sous-réseau doit être configuré comme décrit dans Configuration de la mise en réseau (p. 137). En outre, vérifiez que le sous-réseau dans lequel vous lancez le cluster dispose de suffisamment d'adresses IP Elastic libres pour en attribuer une à chaque nœud du cluster.

Step 3: Look at the Last State Change

Le dernier changement d'état fournit des informations sur ce qui s'est passé la dernière fois que le cluster a changé d'état. Cela contient souvent des informations qui peuvent vous indiquer ce qui s’est mal passé car un cluster change de statut à `FAILED`. Par exemple, si vous lancez un cluster de diffusion en continu et spécifiez un emplacement de sortie qui existe déjà dans Amazon S3, le cluster échoue avec un dernier changement d’état de « Le répertoire de sortie en continu existe déjà ».

Vous pouvez localiser la valeur du dernier changement d'état dans la console en affichant le volet de détails pour le cluster, à partir de l'interface de ligne de commande à l'aide des arguments `list-steps` ou `describe-cluster`, ou à partir de l'API à l'aide des actions `DescribeCluster` et `ListSteps`. Pour plus d'informations, consultez View Cluster Status and Details (p. 319),

Step 4: Examine the Log Files

L'étape suivante consiste à examiner les fichiers journaux afin de rechercher un code d'erreur ou toute autre indication du problème rencontré par votre cluster. Pour plus d'informations sur les fichiers journaux disponibles, leur emplacement et leur mode d'affichage, consultez la page View Log Files (p. 333).

Vous risquez de devoir effectuer quelques recherches pour déterminer ce qui s'est passé. Hadoop exécute le travail des tâches dans des tentatives de tâche sur différents nœuds du cluster. Amazon EMR peut lancer des tentatives de tâches spéculatives, mettant d'abord fin aux autres tentatives de tâches qui ne sont pas terminées. Cela génère une activité importante qui est consignée au fur et à mesure dans

les fichiers journaux contrôleur, stderr et système. En outre, plusieurs tentatives de tâche s'exécutent simultanément, mais un fichier journal ne peut afficher les résultats qu'en mode linéaire.

Commencez par rechercher dans les journaux d'action d'amorçage les erreurs ou les changements de configuration inattendus lors du lancement du cluster. Consultez ensuite les journaux d'étape pour identifier les tâches Hadoop lancées dans le cadre d'une étape présentant des erreurs. Examinez les journaux de tâche Hadoop pour identifier les tentatives de tâche qui ont échoué. Le journal des tentatives de tâche contient des détails relatifs à la raison de l'échec de la tentative de tâche.

Les sections suivantes expliquent comment utiliser les différents fichiers journaux pour identifier l'erreur dans votre cluster.

Check the Bootstrap Action Logs

Les actions d'amorçage exécutent des scripts sur le cluster lors de son lancement. Elles sont généralement utilisées pour installer des logiciels supplémentaires sur le cluster ou pour remplacer les valeurs par défaut des paramètres de configuration. La consultation de ces journaux peut fournir des détails sur les erreurs qui se sont produites lors de la configuration du cluster, ainsi que sur les modifications des paramètres de configuration qui pourraient affecter les performances.

Check the Step Logs

Il existe quatre types de journaux d'étape.

- controller—Contient les fichiers générés par Amazon EMR (Amazon EMR) qui proviennent des erreurs rencontrées lors de la tentative d'exécution de l'étape. Si l'étape échoue lors de son chargement, vous pouvez trouver la trace de la pile dans ce journal. Les erreurs de chargement ou d'accès à votre application sont souvent décrites ici, tout comme les erreurs de fichier de mappage manquant.
- stderr—Contient les messages d'erreurs qui se sont produites pendant le traitement de l'étape. Les erreurs de chargement de l'application sont souvent décrites ici. Ce journal contient parfois une trace de la pile.
- stdout— contient le statut généré par vos exécutables de mappage et de réduction. Les erreurs de chargement de l'application sont souvent décrites ici. Ce journal contient parfois des messages d'erreur d'application.
- syslog—Contient des journaux de logiciels non Amazon, comme Apache et Hadoop. Les erreurs de streaming sont souvent décrites ici.

Recherchez les erreurs évidentes dans le journal stderr. Si stderr affiche une courte liste d'erreurs, l'étape s'est rapidement arrêtée suite au renvoi d'une erreur. Cela est le plus souvent dû à une erreur dans les applications de mappage et de réduction exécutées dans le cluster.

Examinez les dernières lignes du contrôleur et du journal système pour consulter les avis d'erreur ou de défaillance. Suivez les avis sur les tâches ayant échoué, en particulier s'ils indiquent « Tâche ayant échoué ».

Check the Task Attempt Logs

Si l'analyse précédente des journaux d'étape a révélé l'échec d'une ou plusieurs tâches, consultez les journaux des tentatives de tâche correspondantes pour obtenir plus d'informations sur l'erreur.

Step 5: Test the Cluster Step by Step

Une technique utile lorsque vous essayez de déceler la source d'une erreur consiste à redémarrer le cluster et à lui soumettre les étapes une par une. Cela vous permet de vérifier les résultats de chaque étape avant de traiter la suivante, et vous donne l'occasion de corriger et de réexécuter une étape qui a échoué. Cela présente également l'avantage de vous faire charger une seule fois les données d'entrée.

Pour tester un cluster étape par étape

1. Lancez un nouveau cluster avec les deux options keep-alive et protection contre l'arrêt activées. L'option keep-alive maintient le cluster actif après qu'il a traité toutes ses étapes en suspens. La protection contre l'arrêt empêche un cluster de s'arrêter en cas d'erreur. Pour de plus amples informations, veuillez consulter Configuring a Cluster to Auto-Terminate or Continue (p. 110) et Using Termination Protection (p. 111).
2. Soumettez une étape au cluster. Pour plus d'informations, consultez Submit Work to a Cluster (p. 418),
3. A la fin du traitement de l'étape, recherchez les erreurs dans les fichiers journaux d'étape. Pour plus d'informations, consultez Step 4: Examine the Log Files (p. 431), Le moyen le plus rapide de localiser ces fichiers journaux est de se connecter au nœud maître et d'y afficher les fichiers journaux. Les fichiers journaux d'étape n'apparaissent pas tant que l'étape ne s'est pas exécutée assez longtemps, ne s'est pas terminée ou n'a pas échoué.
4. Si l'étape a réussi sans erreur, exécutez l'étape suivante. Si des erreurs se sont produites, enquêtez sur l'erreur dans les fichiers journaux. Si l'erreur se situe dans votre code, effectuez la correction et réexécutez l'étape. Continuez jusqu'à ce que toutes les étapes s'exécutent sans erreur.
5. Lorsque vous avez terminé le débogage du cluster et souhaitez arrêter ce dernier, vous devez l'arrêter manuellement. Cela est nécessaire car le cluster a été lancé avec la protection contre l'arrêt activée. Pour plus d'informations, consultez Using Termination Protection (p. 111),

Résolution des problèmes de rapidité d'un cluster

Cette section explique le processus de résolution des problèmes d'un cluster qui est en cours d'exécution, mais qui met longtemps à renvoyer les résultats. Pour plus d'informations sur les actions à mettre en œuvre si un code d'erreur a été émis lors de la mise hors service du cluster, consultez Troubleshoot a Failed Cluster (p. 429)

Amazon EMR vous permet de spécifier le nombre et le type d'instances dans le cluster. Ces éléments sont ceux qui ont le plus grand effet sur la vitesse de traitement de vos données. Vous pouvez envisager de ré-exécuter le cluster, cette fois-ci en spécifiant les instances EC2 avec davantage de ressources, ou en spécifiant un plus grand nombre d'instances dans le cluster. Pour plus d'informations, consultez Configuration du matériel et de la mise en réseau d'un cluster (p. 129).

Les rubriques suivantes vous expliquent comment identifier d'autres causes possibles du ralentissement d'un cluster.

Rubriques

- Étape 1. Recueillir des données sur le problème (p. 433)
- Étape 2. Vérifier l’environnement (p. 434)
- Étape 3 Examiner les fichiers journaux (p. 435)
- Étape 4. Vérifier la santé des groupes et des instances (p. 436)
- Étape 5 : Vérifier les groupes suspendus (p. 437)
- Étape 6 : Vérifier les paramètres de configuration (p. 438)
- Étape 7. Examiner les données d’entrée (p. 439)

Étape 1. Recueillir des données sur le problème

La première étape de résolution des problèmes d'un cluster consiste à collecter des informations sur l'incident, ainsi que sur l'état et la configuration actuels du cluster. Ces informations seront utilisées dans les étapes suivantes pour confirmer ou infirmer les causes possibles du problème.

Définition du problème

Pour commencer, définissez clairement le problème. Il convient de vous poser quelques questions :

- Quel était le résultat attendu ? Que s'est-il passé à la place ?
- Quand ce problème s'est-il produit pour la première fois ? A quelle fréquence s'est-il produit depuis ?
- Ai-je changé quelque chose dans la configuration ou l'exécution de mon cluster ?

Détails du cluster

Les détails de cluster suivants sont utiles pour aider à localiser les problèmes. Pour en savoir plus sur la collecte de ces informations, consultez la page View Cluster Status and Details (p. 319).

- Identificateur du cluster (également appelé identificateur de flux de travail).
- Région et zone de disponibilité où le cluster a été lancé.
- État du cluster, notamment les détails du dernier changement d'état.
- Type et nombre d'instances EC2 spécifiés pour les nœuds maîtres, principaux et de tâches.

Étape 2. Vérifier l’environnement

Rubriques

- Recherche d'interruptions de service (p. 434)
- Recherche des limites d'utilisation (p. 434)
- Vérifiez le Amazon VPC Configuration du sous-réseau (p. 435)
- Redémarrage du cluster (p. 435)

Recherche d'interruptions de service

Amazon EMR utilise plusieurs Amazon Web Services en interne. Il exécute des serveurs virtuels sur Amazon EC2, stocke les données et les scripts sur Amazon S3, indexe les fichiers journaux dans Amazon SimpleDB et communique les métriques à CloudWatch. Les événements qui perturbent ces services sont rares mais, lorsqu'ils se produisent, peuvent provoquer des problèmes dans Amazon EMR.

Avant d'aller plus loin, consultez le Tableau de bord de l'état des services. Examinez la région où vous avez lancé votre cluster pour voir s'il existe des événements de perturbation dans l'un de ces services.

Recherche des limites d'utilisation

Si vous lancez un cluster volumineux et que vous avez lancé de nombreux clusters simultanément, ou si vous êtes un utilisateur IAM partageant un compte AWS avec d'autres utilisateurs, le cluster est peut-être tombé en panne parce que vous avez dépassé une limite de service AWS.

Amazon EC2 limite le nombre d'instances de serveur virtuel en cours d'exécution sur une seule région AWS à 20 instances réservées ou à la demande. Si vous lancez un cluster avec plus de 20 nœuds, ou que vous lancez un cluster qui entraîne le dépassement du nombre total d'instances EC2 actives sur votre compte AWS au-delà de 20, le cluster ne pourra pas lancer toutes les instances EC2 nécessaires et risque de tomber en panne. Dans ce cas, Amazon EMR renvoie une erreur `EC2 QUOTA EXCEEDED`. Vous pouvez demander à ce qu'AWS augmente le nombre d'instances EC2 que vous pouvez exécuter sur votre compte en envoyant une demande d'augmentation de la limite d'instances Amazon EC2.

Une autre raison pouvant provoquer le dépassement de vos limites d'utilisation est le délai entre l'arrêt d'un cluster et la libération de la totalité de ses ressources. En fonction de sa configuration, l'arrêt complet d'un cluster et la libération des ressources allouées peut prendre de 5 à 20 minutes. Si vous obtenez une erreur `EC2 QUOTA EXCEEDED` lorsque vous tentez de lancer un cluster, il est possible que certaines ressources provenant d'un cluster récemment arrêté n'aient pas encore été libérées. Dans ce cas, vous pouvez soit demander à ce que votre quota Amazon EC2 soit augmenté, soit attendre 20 minutes et relancer le cluster.

Amazon S3 limite le nombre de compartiments créés sur un compte à 100. Si votre cluster crée un compartiment qui dépasse cette limite, la création du compartiment échoue et peut provoquer l'arrêt du cluster.

Vérifiez le Amazon VPC Configuration du sous-réseau

Si votre cluster a été lancé dans un sous-réseau Amazon VPC, le sous-réseau doit être configuré comme décrit dans Configuration de la mise en réseau (p. 137). En outre, vérifiez que le sous-réseau dans lequel vous lancez le cluster dispose de suffisamment d'adresses IP Elastic libres pour en attribuer une à chaque nœud du cluster.

Redémarrage du cluster

Une condition temporaire peut être à l'origine du ralentissement du traitement. Envisagez d'arrêter et de redémarrer le cluster pour vérifier si cela permet d'améliorer les performances.

Étape 3 Examiner les fichiers journaux

L'étape suivante consiste à examiner les fichiers journaux afin de rechercher un code d'erreur ou toute autre indication du problème rencontré par votre cluster. Pour plus d'informations sur les fichiers journaux disponibles, leur emplacement et leur mode d'affichage, consultez la page View Log Files (p. 333).

Vous risquez de devoir effectuer quelques recherches pour déterminer ce qui s'est passé. Hadoop exécute le travail des tâches dans des tentatives de tâche sur différents nœuds du cluster. Amazon EMR peut lancer des tentatives de tâches spéculatives, mettant d'abord fin aux autres tentatives de tâches qui ne sont pas terminées. Cela génère une activité importante qui est consignée au fur et à mesure dans les fichiers journaux contrôleur, stderr et système. En outre, plusieurs tentatives de tâche s'exécutent simultanément, mais un fichier journal ne peut afficher les résultats qu'en mode linéaire.

Commencez par rechercher dans les journaux d'action d'amorçage les erreurs ou les changements de configuration inattendus lors du lancement du cluster. Consultez ensuite les journaux d'étape pour identifier les tâches Hadoop lancées dans le cadre d'une étape présentant des erreurs. Examinez les journaux de tâche Hadoop pour identifier les tentatives de tâche qui ont échoué. Le journal des tentatives de tâche contient des détails relatifs à la raison de l'échec de la tentative de tâche.

Les sections suivantes expliquent comment utiliser les différents fichiers journaux pour identifier l'erreur dans votre cluster.

Vérification des journaux d'actions d'amorçage

Les actions d'amorçage exécutent des scripts sur le cluster lors de son lancement. Elles sont généralement utilisées pour installer des logiciels supplémentaires sur le cluster ou pour remplacer les valeurs par défaut des paramètres de configuration. La consultation de ces journaux peut fournir des détails sur les erreurs qui se sont produites lors de la configuration du cluster, ainsi que sur les modifications des paramètres de configuration qui pourraient affecter les performances.

Vérification des journaux d'étape

Il existe quatre types de journaux d'étape.

- controller—Contient les fichiers générés par Amazon EMR (Amazon EMR) qui proviennent des erreurs rencontrées lors de la tentative d'exécution de l'étape. Si l'étape échoue lors de son chargement, vous pouvez trouver la trace de la pile dans ce journal. Les erreurs de chargement ou d'accès à votre application sont souvent décrites ici, tout comme les erreurs de fichier de mappage manquant.
- stderr—Contient les messages d'erreurs qui se sont produites pendant le traitement de l'étape. Les erreurs de chargement de l'application sont souvent décrites ici. Ce journal contient parfois une trace de la pile.
- stdout— contient le statut généré par vos exécutables de mappage et de réduction. Les erreurs de chargement de l'application sont souvent décrites ici. Ce journal contient parfois des messages d'erreur d'application.
- syslog—Contient des journaux de logiciels non Amazon, comme Apache et Hadoop. Les erreurs de streaming sont souvent décrites ici.

Recherchez les erreurs évidentes dans le journal stderr. Si stderr affiche une courte liste d'erreurs, l'étape s'est rapidement arrêtée suite au renvoi d'une erreur. Cela est le plus souvent dû à une erreur dans les applications de mappage et de réduction exécutées dans le cluster.

Examinez les dernières lignes du contrôleur et du journal système pour consulter les avis d'erreur ou de défaillance. Suivez les avis sur les tâches ayant échoué, en particulier s'ils indiquent « Tâche ayant échoué ».

Vérification des journaux de tentatives de tâche

Si l'analyse précédente des journaux d'étape a révélé l'échec d'une ou plusieurs tâches, consultez les journaux des tentatives de tâche correspondantes pour obtenir plus d'informations sur l'erreur.

Vérification des journaux des démons Hadoop

Dans de rares cas, Hadoop lui-même peut échouer. Pour voir si tel est le cas, vous devez consulter les journaux Hadoop. Il est situé sous `/var/log/hadoop/` sur chaque nœud.

Vous pouvez utiliser les journaux JobTracker pour mapper une tentative de tâche ayant échoué vers le nœud sur lequel elle était exécutée. Une fois que vous connaissez le nœud associé à la tentative de tâche, vous pouvez vérifier l'état de santé de l'instance EC2 hébergeant ce nœud pour voir si des problèmes tels qu'un nombre insuffisant d'UC ou une saturation de la mémoire se sont produits.

Étape 4. Vérifier la santé des groupes et des instances

Un Amazon EMR le cluster est constitué de nœuds fonctionnant sur Amazon EC2 instances. Si ces instances deviennent dépendantes des ressources (par exemple, si l'UC ou la mémoire est saturée), rencontrent des problèmes de connectivité réseau ou sont mises hors service, cela a un impact sur la vitesse de traitement du cluster.

Il existe jusqu'à trois types de nœuds dans un cluster :

- nœud maître — gère le cluster. En cas de problème de performances, l'ensemble du cluster est affecté.
- nœuds centraux — la carte des processus-réduire les tâches et maintenir le système de fichiers distribué Hadoop (HDFS). Si l'un de ces nœuds rencontre des problèmes de performances, cela peut ralentir les opérations du système de fichiers distribué Hadoop ainsi que le traitement MapReduce. Vous pouvez ajouter des nœuds principaux supplémentaires à un cluster pour améliorer les performances, mais vous ne pouvez pas supprimer les nœuds principaux. Pour plus d'informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution (p. 410).
- nœuds de tâches — processus de réduction de carte de processus. Il s'agit de ressources de calcul uniquement. Ils ne stockent pas de données. Vous pouvez ajouter des nœuds de tâches à un cluster

pour accélérer les performances, ou supprimer les nœuds de tâches qui sont inutiles. Pour plus d'informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution (p. 410).

Lorsque vous vérifiez l'état d'un cluster, vous devez prendre en compte les performances du cluster dans son ensemble, ainsi que les performances des instances individuelles. Vous pouvez utiliser plusieurs outils :

Vérification de l'état du cluster avec CloudWatch

Chaque cluster Amazon EMR communique des métriques à CloudWatch. Ces métriques fournissent des informations résumées sur les performances du cluster, telles que la charge totale, l'utilisation HDFS, les tâches en cours d'exécution, les tâches restantes, les blocs corrompus etc. Les métriques CloudWatch vous offrent une vue d'ensemble de votre cluster et peuvent aider à comprendre l'origine du ralentissement du traitement. En plus d'utiliser CloudWatch pour analyser un problème de performance existant, vous pouvez définir des alarmes qui provoquent CloudWatch pour alerter si un problème de performance futur survient. Pour plus d'informations, consultez Surveillance des métriques avec CloudWatch (p. 348).

Vérifier le statut du travail et l'état HDFS

Utilisez le Interfaces utilisateur d'application sur la page des détails du cluster pour afficher les détails de l'application FIL. Pour certaines applications, vous pouvez explorer plus en détail et accéder aux journaux directement. Cette fonctionnalité est particulièrement utile pour les applications Spark. Pour plus d'informations, consultez Afficher l'historique de l'application (p. 327).

Hadoop offre une série d'interfaces web que vous pouvez utiliser pour afficher des informations. Pour plus d'informations sur la façon d'accéder à ces interfaces web, consultez Affichage des interfaces web hébergées sur des clusters Amazon EMR (p. 372).

- JobTracker — fournit des informations sur la progression de la tâche en cours de traitement par le cluster. Vous pouvez utiliser cette interface pour savoir quand un travail se bloque.
- HDFS NameNode — fournit des informations sur le pourcentage d'utilisation HDFS et l'espace disponible sur chaque nœud. Vous pouvez utiliser cette interface pour savoir quand HDFS devient dépendant des ressources et nécessite une capacité supplémentaire.
- TaskTracker — fournit des informations sur les tâches du travail en cours de traitement par le cluster. Vous pouvez utiliser cette interface pour savoir quand une tâche se bloque.

Vérification de l'état de l'instance avec Amazon EC2

La console Amazon EC2 permet également de rechercher des informations sur l'état des instances de votre cluster. Etant donné que chaque nœud du cluster s'exécute sur une instance EC2, vous pouvez utiliser les outils fournis par Amazon EC2 pour vérifier leur état. Pour plus d'informations, consultez View Cluster Instances in Amazon EC2 (p. 337).

Étape 5 : Vérifier les groupes suspendus

Un groupe d'instances se met en suspension lorsqu'il rencontre trop d'erreurs lors de la tentative de lancement des nœuds. Par exemple, si de nouveaux nœuds échouent plusieurs fois lors de l'exécution des actions d'amorçage, le groupe d'instances — après un certain temps — allez dans le `SUSPENDED` plutôt que de tenter continuellement de provisionner de nouveaux nœuds.

Le traitement d'un nœud peut échouer si :

- Hadoop ou le cluster est endommagé et n'accepte pas de nouveau nœud dans le cluster

- Une action d'amorçage échoue sur le nouveau nœud
- Le nœud ne fonctionne pas correctement et sa vérification échoue avec Hadoop

Si un groupe d'instances est dans le `SUSPENDED` et le cluster est dans un `WAITING` état, vous pouvez ajouter une étape cluster pour réinitialiser le nombre souhaité de nœuds de noyau et de tâche. L'ajout de l'étape reprend le traitement du cluster et met le groupe d'instances dans un `RUNNING` état.

Pour plus d'informations sur la réinitialisation d'un cluster dans un état suspendu, voir État suspendu (p. 413).

Étape 6 : Vérifier les paramètres de configuration

Les paramètres de configuration spécifient les informations relatives à l'exécution d'un cluster, comme le nombre de nouvelles tentatives pour une tâche et la quantité de mémoire disponible pour le tri. Lorsque vous lancez un cluster en utilisant Amazon EMR, il y a Amazon EMR-paramètres spécifiques en plus des paramètres de configuration Hadoop standard. Les paramètres de configuration sont stockés dans le nœud maître du cluster. Vous pouvez vérifier les paramètres de configuration pour vous assurer que votre cluster possède les ressources nécessaires pour s'exécuter de manière efficace.

Amazon EMR définit les paramètres de configuration par défaut Hadoop qu'il utilise pour lancer un cluster. Les valeurs sont basées sur l'AMI et le type d'instance que vous spécifiez pour le cluster. Vous pouvez modifier les valeurs par défaut des paramètres de configuration à l'aide d'une action d'amorçage ou en spécifiant les nouvelles valeurs dans les paramètres de l'exécution des tâches. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125). Pour déterminer si une action d'amorçage a modifié les paramètres de configuration, vérifiez les journaux des actions d'amorçage.

Amazon EMR enregistre les paramètres de Hadoop utilisés pour exécuter chaque tâche. Les données du journal sont stockées dans un fichier nommé `job_`*`job-id`*`_conf.xml` sous le `/mnt/var/log/hadoop/history/` répertoire du nœud principal, où *`job-id`* est remplacé par l'identifiant du poste. Si vous avez activé l'archivage des journaux, ces données sont copiées vers Amazon S3 dans le `logs/`*`date`*`/`*`jobflow-id`*`/jobs` dossier, où *`date`* est la date à laquelle le travail a été exécuté, et *`jobflow-id`* est l'identifiant du cluster.

Les paramètres de configuration des tâches Hadoop suivants sont particulièrement utiles pour tenter de résoudre les problèmes de performances. Pour plus d'informations sur les paramètres de configuration Hadoop et sur la manière dont ils affectent le comportement de Hadoop, allez sur http://hadoop.apache.org/docs/.

Paramètre de configuration	Description :
dfs.replication	Nombre de nœuds HDFS dans lesquels un seul bloc (par exemple, le bloc disque dur) est copié afin de produire un environnement de type RAID. Détermine le nombre de nœuds HDFS contenant une copie du bloc.
io.sort.mb	Quantité totale de mémoire disponible pour le tri. Cette valeur doit être 10x io.sort.factor. Ce paramètre peut aussi être utilisé pour calculer la mémoire totale utilisée par chaque nœud de tâche, par l'opération io.sort.mb multiplié par mapred.tasktracker.ap.tasks.maximum.
io.sort.spill.percent	Utilisé pendant le tri. Point auquel le disque commence à être utilisé car la mémoire allouée pour le tri est saturée.
mapred.child.java.opts	(Obsolète) Utiliser mapred.map.child.java.opts et mapred.reduce.child.java.opts à la place. Options Java utilisées par TaskTracker lors du lancement d'une machine

Paramètre de configuration	Description :
	virtuelle Java afin d'y exécuter une tâche. « -Xmx » est un paramètre courant pour définir la taille maximale de la mémoire.
mapred.map.child.java.opts	Options Java utilisées par TaskTracker lors du lancement d'une machine virtuelle Java afin d'y exécuter une tâche de mappage. « -Xmx » est un paramètre courant pour définir la taille maximale du tas de la mémoire.
mapred.map.tasks.speculative.execution	Détermine si les tentatives de tâches de mappage de la même tâche peuvent être lancées en parallèle.
mapred.reduce.tasks.speculative.execution	Détermine si les tentatives de tâches de réduction de la même tâche peuvent être lancées en parallèle.
mapred.map.max.Attempts	Nombre maximum de tentatives pour une tâche de mappage. Si toutes les tentatives échouent, la tâche de mappage est marquée comme ayant échoué.
mapred.reduce.child.java.opts	Options Java utilisées par TaskTracker lors du lancement d'une machine virtuelle Java afin d'y exécuter une tâche de réduction. « -Xmx » est un paramètre courant pour définir la taille maximale du tas de la mémoire.
mapred.reduce.max.attempts	Nombre maximum de tentatives pour une tâche de réduction. Si toutes les tentatives échouent, la tâche de mappage est marquée comme ayant échoué.
mapred.reduce.slowstart.completed.maps	Quantité de tâches de mappage devant se terminer avant le lancement de tâches de réduction. Si le temps d'attente est trop court, des erreurs « Too many fetch-failure » peuvent survenir lors des tentatives.
mapred.reuse.jvm.num.tasks	Une tâche s'exécute dans une même machine virtuelle Java. Spécifie le nombre de tâches pouvant réutiliser la même machine virtuelle Java.
mapred.tasktracker.map.tasks.maximum	Quantité maximale de tâches qui peuvent s'exécuter en parallèle par nœud de tâches au cours du mappage.
mapred.tasktracker.reduce.tasks.maximum	Quantité maximale de tâches qui peuvent s'exécuter en parallèle par nœud de tâches au cours de la réduction.

Si vos tâches de cluster utilisent beaucoup de mémoire, vous pouvez améliorer les performances en utilisant un nombre inférieur de tâches par nœud principal et en réduisant la taille du tas de votre dispositif de suivi des travaux.

Étape 7. Examiner les données d’entrée

Observez vos données d'entrée. Sont-elles réparties de manière uniforme sur vos valeurs de clés ? Si vos données sont majoritairement réparties vers une ou seulement quelques valeurs clés, la charge de traitement peut être mappée à un petit nombre de nœuds alors que d'autres nœuds sont inutilisés. Cette distribution déséquilibrée du travail peut entraîner un ralentissement de traitement.

Voici un exemple d'ensemble de données déséquilibré : un cluster est exécuté pour trier des mots par ordre alphabétique, mais l'ensemble de données contient uniquement des mots commençant par la lettre

« a ». Le nœud qui traite les valeurs commençant par « a » est surchargé, tandis que les nœuds qui traitent les mots commençant par d'autres lettres sont inactifs.

Common Errors in Amazon EMR

Il existe de nombreuses raisons pour lesquelles un cluster peut échouer ou être lent dans le traitement des données. Les sections suivantes répertorient les problèmes et les suggestions les plus courants pour les corriger.

Rubriques

- Input and Output Errors (p. 440)
- Erreurs d'autorisations (p. 442)
- Erreurs de ressource (p. 443)
- Streaming Cluster Errors (p. 450)
- Custom JAR Cluster Errors (p. 451)
- Hive Cluster Errors (p. 452)
- VPC Errors (p. 453)
- AWS GovCloud (US-West) Errors (p. 456)
- Other Issues (p. 456)

Input and Output Errors

Les erreurs suivantes sont courantes dans les opérations d'entrée et sortie de cluster.

Rubriques

- Does your path to Amazon Simple Storage Service (Amazon S3) have at least three slashes? (p. 440)
- Are you trying to recursively traverse input directories? (p. 440)
- Does your output directory already exist? (p. 441)
- Are you trying to specify a resource using an HTTP URL? (p. 441)
- Are you referencing an Amazon S3 bucket using an invalid name format? (p. 441)
- Are you experiencing trouble loading data to or from Amazon S3? (p. 441)

Does your path to Amazon Simple Storage Service (Amazon S3) have at least three slashes?

Lorsque vous spécifiez un compartiment Amazon S3, vous devez inclure une barre oblique de terminaison à la fin de l'URL. Par exemple, au lieu de référencer un compartiment sous la forme « s3n://*DOC-EXAMPLE-BUCKET1* », vous devez utiliser « s3n:///*DOC-EXAMPLE-BUCKET1* », sinon Hadoop ne prendra pas en compte votre cluster dans la plupart des cas.

Are you trying to recursively traverse input directories?

Hadoop ne recherche pas de façon récursive les fichiers dans les répertoires d'entrée. Si vous avez une structure de répertoire telle que /corpus/01/01.txt, /corpus/01/02.txt, /corpus/02/01.txt, etc., et que vous spécifiez /corpus/ comme paramètre d'entrée pour votre cluster, Hadoop ne trouve aucun fichier d'entrée,

car le répertoire /corpus/ est vide et Hadoop ne vérifie pas le contenu des sous-répertoires. De même, Hadoop ne vérifie pas de façon récursive les sous-répertoires des compartiments Amazon S3.

Les fichiers d'entrée doivent être directement dans le répertoire d'entrée ou le compartiment Amazon S3 que vous spécifiez, pas dans les sous-répertoires.

Does your output directory already exist?

Si vous spécifiez un chemin de sortie qui existe déjà, Hadoop entraîne dans la plupart des cas l'échec du cluster. Cela signifie que si vous exécutez un cluster une première fois, puis l'exécutez à nouveau avec exactement les mêmes paramètres, il fonctionnera probablement la première fois puis jamais plus. Après la première utilisation, le chemin de sortie existe et il entraîne l'échec de toutes les exécutions suivantes.

Are you trying to specify a resource using an HTTP URL?

Hadoop n'accepte pas les emplacements de ressources utilisant le préfixe http://. Vous ne pouvez pas faire référence à une ressource à l'aide d'une URL HTTP. Par exemple, transmettre http://mysite/myjar.jar comme paramètre JAR provoque l'échec du cluster.

Are you referencing an Amazon S3 bucket using an invalid name format?

Si vous essayez d'utiliser un nom de compartiment tel que « *DOC-EXAMPLE-BUCKET1*.1 » avec Amazon EMR, votre cluster échoue car Amazon EMR exige que les noms de compartiments soient des noms d'hôtes RFC 2396 valides ; le nom ne peut pas se terminer par un chiffre. En outre, en raison des exigences de Hadoop, les noms des compartiments Amazon S3 utilisés avec Amazon EMR doivent contenir uniquement des lettres minuscules, des chiffres, des points (.) et des traits d'union (-). Pour de plus amples informations sur la façon de former les noms des compartiments Amazon S3, veuillez consulter Limites et restrictions applicables aux compartiments dans le Amazon Simple Storage Service Manuel du développeur.

Are you experiencing trouble loading data to or from Amazon S3?

Amazon S3 est la source d'entrée et de sortie la plus populaire pour Amazon EMR. Une erreur courante est de traiter Amazon S3 comme un système de fichiers standard. Il existe des différences entre Amazon S3 et un système de fichiers que vous devez prendre en compte lorsque vous exécutez votre cluster.

- If an internal error occurs in Amazon S3, your application needs to handle this gracefully and re-try the operation.
- If calls to Amazon S3 take too long to return, your application may need to reduce the frequency at which it calls Amazon S3.
- Listing all the objects in an Amazon S3 bucket is an expensive call. Your application should minimize the number of times it does this.

Il existe plusieurs façons d'améliorer la façon dont votre cluster interagit avec Amazon S3.

- Launch your cluster using the most recent release version of Amazon EMR.
- Use S3DistCp to move objects in and out of Amazon S3. S3DistCp implements error handling, retries and back-offs to match the requirements of Amazon S3. For more information, see Distributed Copy Using S3DistCp.
- Design your application with eventual consistency in mind. Use HDFS for intermediate data storage while the cluster is running and Amazon S3 only to input the initial data and output the final results.
- If your clusters will commit 200 or more transactions per second to Amazon S3, contact support to prepare your bucket for greater transactions per second and consider using the key partition strategies described in Amazon S3 Performance Tips & Tricks.

- Set the Hadoop configuration setting io.file.buffer.size to 65536. This causes Hadoop to spend less time seeking through Amazon S3 objects.
- Consider disabling Hadoop's speculative execution feature if your cluster is experiencing Amazon S3 concurrency issues. This is also useful when you are troubleshooting a slow cluster. You do this by setting the `mapreduce.map.speculative` and `mapreduce.reduce.speculative` properties to `false`. When you launch a cluster, you can set these values using the `mapred-env` configuration classification. Pour de plus amples informations, veuillez consulter Configuration des applications dans le Amazon EMR Guide de version.
- If you are running a Hive cluster, see Are you having trouble loading data to or from Amazon S3 into Hive? (p. 452).

Pour plus d'informations, consultez Bonnes pratiques en matière d'erreurs liées à Amazon S3 dans le Amazon Simple Storage Service Manuel du développeur.

Erreurs d'autorisations

Les erreurs suivantes sont courantes lors de l'utilisation des autorisations ou des informations d'identification.

Rubriques

- Les informations d'identification que vous saisissez dans SSH sont-elles correctes ? (p. 442)
- Si vous utilisez IAM, les stratégies Amazon EC2 appropriées sont-elles définies ? (p. 443)

Les informations d'identification que vous saisissez dans SSH sont-elles correctes ?

Si vous ne pouvez pas utiliser SSH pour vous connecter au nœud maître, il s'agit probablement d'un problème lié à vos informations d'identification de sécurité.

Commencez par vérifier que le fichier .pem qui contient votre clé SSH dispose des autorisations appropriées. Vous pouvez utiliser chmod pour modifier les autorisations sur votre fichier .pem, comme indiqué dans l'exemple suivant, où vous remplacez « mykey.pem » par le nom de votre propre fichier .pem.

```
chmod og-rwx mykey.pem
```

Le problème peut également se produire si vous n'utilisez pas la paire de clés que vous avez spécifiée lors de la création du cluster. C'est une erreur courante si vous avez créé plusieurs paires de clés. Vérifiez les détails du cluster dans la console Amazon EMR (ou utilisez l'option `--describe` dans l'interface de ligne de commande) afin d'identifier le nom de la paire de clés qui a été spécifié lors de la création du cluster.

Après avoir vérifié que vous utilisez la paire de clés correcte et que les autorisations sont définies correctement dans le fichier .pem, vous pouvez utiliser la commande suivante pour utiliser SSH pour vous connecter au nœud principal, où vous remplacez « mykey.pem » par le nom de votre fichier .pem et `hadoop@ec2-01-001-001-1.compute-1.amazonaws.com` par le nom DNS public du nœud principal (disponible via l'option `--describe` dans l'interface de ligne de commande ou via la console Amazon EMR.)

> **Important**
>
> Vous devez utiliser le nom de connexion `hadoop` lorsque vous vous connectez à un nœud de cluster Amazon EMR, sinon une erreur similaire à l'erreur `Server refused our key` peut se produire.

```
ssh -i mykey.pem hadoop@ec2-01-001-001-1.compute-1.amazonaws.com
```

Pour plus d'informations, consultez Connexion au nœud maître à l'aide de SSH (p. 368).

Si vous utilisez IAM, les stratégies Amazon EC2 appropriées sont-elles définies ?

Étant donné qu'Amazon EMR utilise des instances EC2 en tant que nœuds, les utilisateurs IAM d'Amazon EMR ont également besoin de disposer de certaines stratégies Amazon EC2 afin qu'Amazon EMR puisse gérer ces instances pour l'utilisateur IAM. Si les autorisations requises ne sont pas définies, Amazon EMR renvoie l'erreur : "User account is not authorized to call EC2." (« Compte d'utilisateur non autorisé à appeler EC2 »)

Pour de plus amples informations sur les stratégies Amazon EC2 que votre compte IAM doit définir pour exécuter Amazon EMR, veuillez consulter Comment Amazon EMR fonctionne avec IAM (p. 211).

Erreurs de ressource

Les erreurs suivantes sont généralement causées par des ressources limitées sur le cluster.

Rubriques

- Le cluster se termine par NO_SLAVE_LEFT FAILED_BY_MASTER et nœuds principaux FAILED_BY_MASTER (p. 443)
- Cannot replicate block, only managed to replicate to zero nodes (Impossible de répliquer un bloc, réplication sur zéro nœud gérée uniquement) (p. 445)
- EC2 QUOTA EXCEEDED (QUOTA EC2 DÉPASSÉ) (p. 446)
- Too many fetch-failures (Trop d'erreurs de récupération) (p. 446)
- Le fichier a pu uniquement être répliqué sur 0 nœud et non 1 (p. 447)
- Nœuds sur liste noire (p. 448)
- Erreur de Limitation (p. 448)
- Type d'instance non pris en charge (p. 449)
- EC2 is Out of Capacity (EC2 est en manque de capacité) (p. 450)

Le cluster se termine par NO_SLAVE_LEFT FAILED_BY_MASTER et nœuds principaux FAILED_BY_MASTER

Cela se produit généralement en raison de l'arrêt de la protection de la résiliation et tous les nœuds principaux dépassent la capacité de stockage de disque spécifiée par un seuil d'utilisation maximal dans la classification de configuration `yarn-site`, qui correspond au fichier `yarn-site.xml`. Par défaut, cette valeur est 90 %. Lorsque l'utilisation du disque d'un nœud principal dépasse le seuil d'utilisation, le service de santé FIL notariager signale le nœud comme `UNHEALTHY`. Alors que c'est dans cet état, Amazon EMR noire répertorie le nœud et n'attribue pas les conteneurs de FIL à l'emplacement. Si le nœud reste déséquilibré pendant 45 minutes, Amazon EMR marque les Amazon EC2 instance de résiliation `FAILED_BY_MASTER`. Quand tout Amazon EC2 les instances associées aux nœuds centraux sont marquées pour la terminaison, le cluster se termine par le statut `NO_SLAVE_LEFT` car il n'y a pas de ressources pour exécuter les tâches.

Le dépassement de l'utilisation du disque sur un nœud principal peut entraîner une réaction en chaîne. Si un seul nœud dépasse le seuil d'utilisation du disque à cause de HDFS, d'autres nœuds sont également susceptibles d'être proches du seuil. Le premier nœud dépasse le seuil d'utilisation de disque, donc Amazon EMR le met sur la liste noire. Cela augmente la charge de l'utilisation du disque pour les nœuds restants, car ils commenceront à répliquer des données HDFS entre eux qu'ils ont perdues du nœud sur la liste noire. Chaque nœud devient ensuite `UNHEALTHY` de la même manière et le cluster se résilie finalement.

Meilleures pratiques et recommandations

Configuration du matériel de cluster avec un stockage adéquat

Lorsque vous créez un cluster, assurez-vous qu'il y ait suffisamment de nœuds principaux et que chacun possède suffisamment de stockage d'instance et de volumes de stockage EBS pour HDFS. Pour plus d'informations, consultez Calcul de la capacité HDFS requise pour un cluster (p. 166), Vous pouvez également ajouter manuellement des instances principales à des groupes d'instances existants ou en utilisant la mise à l'échelle automatique. Les nouvelles instances possèdent la même configuration de stockage que d'autres instances dans le groupe d'instances. Pour plus d'informations, consultez Dimensionnement des ressources de cluster (p. 384),

Activer la protection de la résiliation

Activer la protection de la résiliation. De cette façon, si un nœud principal est placé sur liste noire, vous pouvez vous connecter à l'instance Amazon EC2 associée à l'aide de SSH pour diagnostiquer les problèmes et récupérer les données. Si vous activez la protection de la résiliation, sachez qu'Amazon EMR ne remplace pas l'instance Amazon EC2 par une nouvelle instance. Pour plus d'informations, consultez Using Termination Protection (p. 111),

Créez une alarme pour la métrique CloudWatch MRUnhealthyNodes

Cette métrique indique le nombre de nœuds de rapports d'un état `UNHEALTHY`. Il est équivalent à la métrique FIL `mapred.resourcemanager.NoOfUnhealthyNodes`. Vous pouvez configurer une notification pour cette alarme pour vous avertir des nœuds non sains avant que le délai de 45 minutes soit atteint. Pour plus d'informations, consultez Surveillance des métriques avec CloudWatch (p. 348),

Affiner les paramètres à l'aide de yarn-site

Les paramètres ci-dessous peuvent être ajustés en fonction des exigences de votre application. Par exemple, vous pouvez augmenter le seuil d'utilisation du disque lorsqu'un nœud signale un état `UNHEALTHY` en augmentant la valeur de `yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage`.

Vous pouvez configurer ces valeurs lorsque vous créez un cluster à l'aide de la classification de configuration `yarn-site`. Pour plus d'informations, consultez Configuration des applications dans le Amazon EMR Guide de version. Vous pouvez également vous connecter aux instances Amazon EC2 associées à des nœuds principaux à l'aide de SSH, puis ajoutez les valeurs dans `/etc/hadoop/conf.empty/yarn-site.xml` à l'aide d'un éditeur de texte. Après avoir apporté la modification, vous devez redémarrer hadoop-yarn-nodemanager comme indiqué ci-dessous.

> **Important**
>
> Lorsque vous redémarrez le service YARN NodeManager, les conteneurs actifs sont supprimés, sauf si `yarn.nodemanager.recovery.enabled` est défini sur `true` à l'aide de la classification de configuration `yarn-site` lorsque vous créez le cluster. Vous devez également spécifier le répertoire dans lequel stocker l'état de conteneur à l'aide de la propriété `yarn.nodemanager.recovery.dir`.

```
sudo /sbin/stop hadoop-yarn-nodemanager
```

```
sudo /sbin/start hadoop-yarn-nodemanager
```

Pour plus d'informations sur les propriétés actuelles de `yarn-site` et les valeurs par défaut, consultez Paramètres YARN par défaut dans la documentation Apache Hadoop.

Propriété	Default Value	Description :
yarn.nodemanager.disk-health-checker.interval-ms	120000	La fréquence (en secondes) à laquelle la vérification de l'état du disque est exécutée.
yarn.nodemanager.disk-health-checker.min-healthy-disks	0,25	La fraction minimale du nombre de disques qui doivent être sains pour que NodeManager lance de nouveaux conteneurs. Cela correspond à la fois à yarn.nodemanager.local-dirs (par défaut, `/mnt/yarn` dans Amazon EMR) et yarn.nodemanager.log-dirs (par défaut `/var/log/hadoop-yarn/containers`, qui est symlinked à `mnt/var/log/hadoop-yarn/containers` dans Amazon EMR).
`yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage`	90.0	Le pourcentage maximal d'utilisation d'espace de disque autorisée après laquelle un disque est marqué comme défectueux. Les valeurs peuvent aller de 0.0 à 100.0. Si la valeur est supérieure ou égale à 100, le NodeManager vérifie si le disque est rempli. Cela s'applique à `yarn-nodemanager.local-dirs` et `yarn.nodemanager.log-dirs`.
`yarn.nodemanager.disk-health-checker.min-free-space-per-disk-mb`	0 %	L'espace minimal qui doit être disponible sur un disque pour qu'il soit utilisé. Cela s'applique à `yarn-nodemanager.local-dirs` et `yarn.nodemanager.log-dirs`.

Cannot replicate block, only managed to replicate to zero nodes (Impossible de répliquer un bloc, réplication sur zéro nœud gérée uniquement)

L’erreur, « Impossible de répliquer le bloc, ne peut être répliquée qu’à zéro noeuds ». survient généralement lorsqu’un cluster n’a pas assez de stockage HDFS. Cette erreur se produit lors de la

génération d'un volume de données dans votre cluster supérieur à ce qui peut être stocké dans HDFS. Vous voyez cette erreur uniquement pendant que le cluster est en cours d'exécution, parce que lorsque la tâche s'arrête, elle libère l'espace HDFS qu'elle utilisait.

La quantité d'espace HDFS disponible pour un cluster dépend du nombre et du type d'instances Amazon EC2 qui sont utilisés en tant que nœuds principaux. Les nœuds de tâche ne sont pas utilisés pour le stockage HDFS. Tout l'espace disque sur chaque instance Amazon EC2, y compris les volumes de stockage EBS attachés, est disponible pour HDFS. Pour de plus amples informations sur la quantité de stockage local pour chaque type d'instance EC2, veuillez consulter Types et familles d'instances dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

L'autre facteur qui peut influer sur la quantité d'espace HDFS disponible est le facteur de réplication, qui correspond au nombre de copies de chaque bloc de données stockées dans HDFS pour la redondance. Le facteur de réplication augmente avec le nombre de nœuds dans le cluster : il y a 3 copies de chaque bloc de données pour un cluster avec 10 nœuds ou plus, 2 copies de chaque bloc pour un cluster avec 4 à 9 nœuds et 1 copie (pas de redondance) pour les clusters avec 3 nœuds ou moins. L'espace total HDFS disponible est divisé par le facteur de réplication. Dans certains cas, tels que l'augmentation du nombre de nœuds de 9 à 10, l'augmentation du facteur de réplication peut effectivement entraîner la diminution de la quantité d'espace HDFS disponible.

Par exemple, un cluster avec dix nœuds principaux de type m1.large auraient 2 833 Go d'espace disponible pour HDFS -((10 nœuds x 850 Go par nœud)/facteur de réplication de 3).

Si votre cluster dépasse la quantité d'espace disponible pour HDFS, vous pouvez ajouter des nœuds principaux supplémentaires à votre cluster ou utiliser des données de compression pour créer davantage d'espace HDFS. Si votre cluster est une version qui peut être arrêtée et redémarrée, vous pouvez envisage d'utiliser des nœuds principaux d'un type d'instance Amazon EC2 plus grand. Vous pouvez également envisager d'ajuster le facteur de réplication. Soyez conscient, cependant, que diminuer le facteur de réplication réduit la redondance des données HDFS et la capacité de votre cluster à récupérer à partir de blocs HDFS perdus ou corrompus.

EC2 QUOTA EXCEEDED (QUOTA EC2 DÉPASSÉ)

Si vous obtenez un message `EC2 QUOTA EXCEEDED`, cela peut avoir plusieurs causes. En fonction des différences de configuration, l'arrêt et la libération des ressources allouées pour les clusters précédents peut prendre de 5 à 20 minutes. Si vous obtenez une erreur `EC2 QUOTA EXCEEDED` lorsque vous tentez de lancer un cluster, il est possible que des ressources provenant d'un cluster récemment arrêté n'aient pas encore été libérées. Ce message peut également être provoqué par le redimensionnement d'un groupe d'instances ou d'un parc d'instances à une taille cible supérieure au quota d'instances actuel pour le compte. Cela peut se produire manuellement ou automatiquement via un dimensionnement automatique.

Vous disposez des options suivantes pour résoudre ce problème :

- Créez une demande de support pour demander une augmentation de quota.
- Si un ou plusieurs clusters en cours d'exécution n'ont pas atteint leur capacité, redimensionnez des groupes d'instances ou réduisez les capacités cibles sur les parcs d'instances pour les clusters en cours d'exécution.
- Créez des clusters avec moins d'instances EC2 ou réduisez la capacité cible.

Too many fetch-failures (Trop d'erreurs de récupération)

La présence de messages d'erreur « Too many fetch-failures (Trop d'erreurs de récupération) » ou « Error reading task output (Erreur de lecture de sortie de tâche) » dans les journaux de tentative de tâche ou d'étape indique que la tâche en cours d'exécution dépend du résultat d'une autre tâche. Cela se produit souvent lorsqu'une tâche de réduction est mise en attente d'exécution et requiert le résultat d'une ou de plusieurs tâches d'action mapper et que le résultat n'est pas encore disponible.

Il existe plusieurs raisons pour lesquelles la sortie peut ne pas être disponible :

- La tâche prérequise est toujours en cours de traitement. Il s'agit souvent d'une tâche de mappage.
- Les données peuvent être indisponibles en raison d'une connectivité réseau médiocre si les données se trouvent sur une autre instance.
- Si HDFS est utilisé pour récupérer le résultat, il peut y avoir un problème avec HDFS.

La cause la plus courante de cette erreur est que la tâche précédente est encore en cours de traitement. Cela est particulièrement probable si les erreurs se produisent lorsque les tâches de réduction essaient en premier de s'exécuter. Vous pouvez vérifier si tel est le cas en passant en revue le journal syslog pour l'étape de cluster qui renvoie l'erreur. Si le syslog indique une progression des deux tâches réduire et mapper, cela indique que la phase de réduction a démarré tandis qu'il existe des tâches mapper qui ne sont pas encore terminées.

Une chose à rechercher dans les journaux est un pourcentage de progression de tâche mapper qui atteint 100 % puis revient à une valeur inférieure. Lorsque le pourcentage de tâche mapper est à 100 %, cela ne signifie pas que toutes les tâches mapper sont terminées. Cela signifie simplement que Hadoop exécute toutes les tâches mapper. Si cette valeur retombe en-dessous de 100 %, cela signifie qu'une tâche mapper a échoué et, en fonction de la configuration, Hadoop peut tenter de replanifier la tâche. Si le pourcentage de tâche de carte reste à 100 % dans les journaux, vérifiez les métriques CloudWatch, spécifiquement `RunningMapTasks`, pour vérifier si la tâche de carte est toujours en cours de traitement. Vous pouvez également trouver ces informations à l'aide de l'interface web de Hadoop sur le nœud maître.

Si vous rencontrez ce problème, il existe plusieurs choses que vous pouvez essayer :

- Demandez à la phase de réduction d'attendre plus longtemps avant de démarrer. Vous pouvez le faire en modifiant le paramètre de configuration mapred.reduce.slowstart.completed.maps Hadoop sur une durée plus longue. Pour plus d'informations, consultez Create Bootstrap Actions to Install Additional Software (p. 125),
- Associez le nombre de réductions à la capacité de réduction totale du cluster. Pour cela, vous ajustez le paramètre de configuration mapred.reduce.tasks Hadoop pour la tâche.
- Utilisez un code de classe d'association afin de réduire le nombre de résultats qui doivent être récupérés.
- Vérifiez qu'il n'y a aucun problème avec le service Amazon EC2 qui affecte les performances réseau du cluster. Vous pouvez le faire à l'aide du Tableau de bord de l'état des services.
- Passez en revue les ressources CPU et la mémoire des instances dans votre cluster pour vous assurer que votre traitement des données ne submerge pas les ressources de vos nœuds. Pour plus d'informations, consultez Configuration du matériel et de la mise en réseau d'un cluster (p. 129),
- Vérifiez la version de l'Amazon Machine Image (AMI) utilisée dans votre cluster Amazon EMR. Si la version est 2.3.0 à 2.4.4 incluse, mettez à jour vers une version ultérieure. Les versions AMI dans la plage spécifiée utilisent une version de Jetty qui peut ne pas parvenir à fournir le résultat souhaité de la phase de mappage. L'erreur d'extraction se produit lorsque les réducteurs ne peuvent pas obtenir le résultat de la phase de mappage.

  Jetty est un serveur HTTP open source qui est utilisé pour des communications machine à machine au sein d'un cluster Hadoop.

Le fichier a pu uniquement être répliqué sur 0 nœud et non 1

Lorsqu'un fichier est écrit dans HDFS, il est répliqué sur plusieurs nœuds principaux. Lorsque vous voyez cette erreur, cela signifie que le démon NameNode n'a aucune instance DataNode disponible pour écrire des données dans HDFS. En d'autres termes, la réplication de bloc n'a pas lieu. Cette erreur peut être provoquée par un certain nombre de problèmes :

- Le système de fichiers HDFS peut être venu à manquer d'espace. C'est la cause la plus probable.

- Les instances DataNode peuvent ne pas avoir été disponibles lorsque la tâche a été exécutée.
- Des instances DataNode ont pu être bloquées à partir de la communication avec le nœud maître.
- Des instances dans le groupe d'instance principal peuvent ne pas être disponibles.
- Des autorisations peuvent être manquantes. Par exemple, le démon JobTracker peut ne pas avoir d'autorisations pour créer des informations de dispositif de suivi des travaux.
- Le paramètre d'espace réservé pour une instance DataNode peut être insuffisant. Vérifiez si tel est le cas en contrôlant le paramètre de configuration dfs.datanode.du.reserved.

Pour vérifier si ce problème est dû à un manque d'espace de disque pour HDFS, consultez la métrique `HDFSUtilization` dans CloudWatch. Si cette valeur est trop élevée, vous pouvez ajouter des nœuds principaux supplémentaires au cluster. Si vous avez un cluster dont vous pensez qu'il peut venir à manquer d'espace HDFS, vous pouvez définir une alarme dans CloudWatch pour vous avertir lorsque la valeur de `HDFSUtilization` dépasse un certain niveau. Pour de plus amples informations, veuillez consulter Redimensionnement manuel d'un cluster en cours d'exécution (p. 410) et Surveillance des métriques avec CloudWatch (p. 348).

Si le problème n'était pas que HDFS vienne à manquer d'espace, vérifiez les journaux DataNode, les journaux NameNode et la connectivité réseau pour d'autres problèmes qui auraient pu empêcher HDFS de répliquer des données. Pour plus d'informations, consultez View Log Files (p. 333),

Nœuds sur liste noire

Le démon NodeManager se charge du lancement et de la gestion des conteneurs sur les nœuds principaux et de tâches. Les conteneurs sont alloués au démon NodeManager par le démon ResourceManager qui s'exécute sur le nœud maître. Le démon ResourceManager surveille le nœud NodeManager via une pulsation.

Il existe plusieurs situations dans lesquelles le démon ResourceManager met sur liste noire un nœud NodeManager, en le supprimant du regroupement de nœuds disponibles pour traiter des tâches :

- Si le nœud NodeManager n'a pas envoyé de pulsation au démon ResourceManager dans les 10 dernières minutes (60 000 millisecondes). Cette période de temps peut être configurée à l'aide du paramètre de configuration `yarn.nm.liveness-monitor.expiry-interval-ms`. Pour plus d'informations sur la modification des paramètres de configuration de Yarn, consultez Configuration des applications dans le Amazon EMR Guide de version.
- nodemanante vérifie la santé des disques déterminés par `yarn.nodemanager.local-dirs` et `yarn.nodemanager.log-dirs`. Les vérifications comprennent les autorisations et l'espace disque libre (< 90 %). Si un disque échoue à la vérification, le démon NodeManager arrête d'utiliser ce disque, mais signale toujours l'état du nœud comme sain. Si plusieurs disques échouent à la vérification, le nœud est signalé comme défectueux au démon ResourceManager et de nouveaux conteneurs ne sont pas attribués au nœud.

Le maître d'application peut également exclure un nœud NodeManager s'il comporte plus de trois tâches ayant échoué. Vous pouvez le remplacer par une valeur plus élevée à l'aide du paramètre de configuration `mapreduce.job.maxtaskfailures.per.tracker`. Les autres paramètres de configuration que vous pouvez modifier contrôlent le nombre de fois pour tenter une tâche avant de la marquer comme ayant échoué : `mapreduce.map.max.attempts` pour les tâches de carte et `mapreduce.reduce.maxattempts` pour réduire les tâches. Pour plus d'informations sur la modification des paramètres de configuration, consultez Configuration des applications dans le Amazon EMR Guide de version.

Erreur de Limitation

Les erreurs « Lancées depuis *`Amazon EC2`* lors du lancement du cluster » et « Échec de la mise à disposition des instances dues à la limitation de *`Amazon EC2`*« se produit lorsque Amazon EMR ne peut

pas terminer une demande car un autre service a déclenché l'activité. Amazon EC2 est la source la plus courante d'erreurs de limitation, mais d'autres services peuvent être la cause d'erreurs de limitation. Les limites de service AWS s'appliquent par région pour améliorer les performances et une erreur de limitation indique que vous avez dépassé la limite de service pour votre compte dans cette région.

Causes possibles

La cause la plus courante d'erreurs de limitation Amazon EC2 est le lancement d'un grand nombre d'instances de cluster entraînant le dépassement de votre limite de service pour les instances EC2. Des instances de cluster peuvent être lancées pour les raisons suivantes :

- De nouveaux clusters sont créés.
- Des clusters sont redimensionnés manuellement. Pour plus d'informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution (p. 410),
- Des groupes d'instances dans un cluster ajoute des instances (montée en charge) en raison d'une règle de dimensionnement automatique. Pour plus d'informations, consultez Présentation des règles de dimensionnement automatique (p. 402),
- Des parcs d'instances dans un cluster ajoutent des instances pour répondre à une augmentation de la capacité cible. Pour plus d'informations, consultez Configuration de parcs d'instances (p. 149),

Il est également possible que la fréquence ou le type de demande d'API faite à Amazon EC2 entraîne des erreurs de limitation. Pour plus d'informations sur la façon dont Amazon EC2 limite les demandes d'API, consultez Débit de demandes d'API de requête dans le Amazon EC2 API Reference.

Solutions

Envisagez les solutions suivantes :

- Créez une demande de support pour demander une augmentation de la limite de service.
- Si des clusters sont lancés selon le même planning (par exemple, au début de l'heure), envisagez d'échelonner les heures de début.
- Si des clusters sont dimensionnés selon les pics de demande et que vous disposez périodiquement d'une capacité d'instance en excès, envisagez de spécifier le dimensionnement automatique pour ajouter et supprimer des instances à la demande. Les instances sont ainsi utilisées de façon plus efficace, et en fonction du profil demande, moins d'instances peuvent être demandées à un moment donné dans un compte. Pour plus d'informations, consultez Utilisation du dimensionnement automatique avec une stratégie personnalisée pour des groupes d'instances (p. 401),

Type d'instance non pris en charge

Si vous créez un cluster, il échoue avec le message d'erreur « Le type d'instance demandée *`InstanceType`* n'est pas pris en charge dans la zone de disponibilité demandée, » signifie que vous avez créé le cluster et spécifié un type d'instance pour un ou plusieurs groupes d'instances qui n'est pas pris en charge par Amazon EMR dans la zone Région et Disponibilité où le cluster a été créé. Amazon EMR peut prendre en charge un type d'instance dans une zone de disponibilité dans une région et non pas une autre. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité au sein de la région.

Solution

Déterminer les types d'instances disponibles dans une zone de disponibilité à l'aide du AWS CLI

- Utilisez la commande `ec2 run-instances` avec l'option `--dry-run`. Dans l'exemple ci-dessous, remplacez *`m5.xlarge`* avec le type d'instance que vous souhaitez utiliser,

ami-035be7bafff33b6b6 avec l'AMI associé à ce type d'instance, et *subnet-12ab3c45* avec un sous-réseau dans la zone de disponibilité que vous souhaitez interroger.

```
aws ec2 run-instances --instance-type m5.xlarge --dry-run --image-
id ami-035be7bafff33b6b6 --subnet-id subnet-12ab3c45
```

Une fois que vous avez déterminé les types d'instance disponibles, vous pouvez effectuer l'une des opérations suivantes :

- Créer le cluster dans la même région et le même sous-réseau EC2, puis choisir un autre type d'instance avec des capacités similaires à celles de votre premier choix. Pour obtenir la liste des types d'instances, consultez Types d'instances pris en charge (p. 132). Pour comparer les capacités des types d'instance EC2, consultez Types d'instance Amazon EC2.
- Choisir un sous-réseau pour le cluster dans une zone de disponibilité où le type d'instance est disponible et pris en charge par Amazon EMR.

EC2 is Out of Capacity (EC2 est en manque de capacité)

Un « EC2 est hors capacité pour *InstanceType*" une erreur se produit lorsque vous tentez de créer un cluster, ou d'ajouter des instances à un cluster dans une zone ou une zone de disponibilité dans laquelle le type d'instance EC2 spécifié est déjà à la capacité. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité.

Pour créer un cluster, procédez comme suit :

- Spécifier un type d'instance différent avec des capacités similaires
- Créer le cluster dans une région différente
- Sélectionnez un sous-réseau dans une zone de disponibilité où le type d'instance que vous souhaitez peut être disponible.

Pour ajouter des instances à un cluster en cours de fonctionnement, procédez comme suit :

- Modifiez les configurations de groupe d'instances ou les configurations de parc d'instances pour ajouter des types d'instances disponibles avec des capacités similaires. Pour obtenir la liste des types d'instances, consultez Types d'instances pris en charge (p. 132). Pour comparer les capacités des types d'instance EC2, consultez Types d'instance Amazon EC2.
- Arrêtez le cluster et recréez-le dans une zone et une zone de disponibilité où le type d'instance est disponible.

Streaming Cluster Errors

Vous pouvez généralement trouver la cause d'une erreur de diffusion en continu dans un fichier `syslog`. Lien vers ce fichier dans le volet Steps (Étapes).

Les erreurs suivantes sont communes aux clusters de diffusion en continu.

Rubriques

- Is data being sent to the mapper in the wrong format? (p. 451)
- Is your script timing out? (p. 451)
- Are you passing in invalid streaming arguments? (p. 451)
- Did your script exit with an error? (p. 451)

Is data being sent to the mapper in the wrong format?

Pour vérifier si tel est le cas, recherchez un message d'erreur dans le fichier `syslog` d'une tentative de tâche ayant échoué dans les journaux de tentative de tâche. Pour plus d'informations, consultez View Log Files (p. 333),

Is your script timing out?

L'expiration par défaut pour un script mappeur ou réducteur est de 600 secondes. Si votre script a besoin de davantage de temps, la tentative de la tâche échoue. Vous pouvez vérifier que c'est le cas en vérifiant le fichier `syslog` d'une tentative de tâche ayant échoué dans les journaux de tentative de tâche. Pour plus d'informations, consultez View Log Files (p. 333),

Vous pouvez modifier le délai en définissant une nouvelle valeur pour le paramètre de configuration `mapred.task.timeout`. Ce paramètre spécifie le nombre de millisecondes après quoi Amazon EMR mettra fin à une tâche qui n'a pas lu l'entrée, écrit la sortie ou mis à jour sa chaîne de statut. Vous pouvez mettre à jour cette valeur en transmettant un argument supplémentaire de diffusion en continu `-jobconf mapred.task.timeout=800000`.

Are you passing in invalid streaming arguments?

La diffusion en continu de Hadoop prend en charge uniquement les arguments suivants. Si vous transmettez des arguments autres que ceux répertoriés ci-après, le cluster échoue.

```
-blockAutoGenerateCacheFiles
-cacheArchive
-cacheFile
-cmdenv
-combiner
-debug
-input
-inputformat
-inputreader
-jobconf
-mapper
-numReduceTasks
-output
-outputformat
-partitioner
-reducer
-verbose
```

En outre, la diffusion en continu Hadoop reconnaît uniquement les arguments transmis à l'aide de la syntaxe Java. C'est à dire, précédés d'un seul trait d'union. Si vous transmettez des arguments précédés d'un tiret double, le cluster échoue.

Did your script exit with an error?

Si votre script mappeur ou réducteur se termine par une erreur, vous pouvez localiser l'erreur dans le fichier `stderr` des journaux de tentative de tâche de la tentative de tâche qui a échoué. Pour plus d'informations, consultez View Log Files (p. 333),

Custom JAR Cluster Errors

Les erreurs suivantes sont communes aux clusters des fichiers JAR personnalisés.

Rubriques

- Is your JAR throwing an exception before creating a job? (p. 452)
- Is your JAR throwing an error inside a map task? (p. 452)

Is your JAR throwing an exception before creating a job?

Si le programme principal de votre fichier JAR personnalisé lève une exception lorsqu'il crée le travail Hadoop, le meilleur emplacement à examiner est le fichier `syslog` des journaux d'étape. Pour plus d'informations, consultez View Log Files (p. 333),

Is your JAR throwing an error inside a map task?

Si votre fichier JAR personnalisé et votre mappeur lèvent une exception lors du traitement de données d'entrée, le meilleur emplacement à examiner est le fichier `syslog` des journaux de tentatives de tâche. Pour plus d'informations, consultez View Log Files (p. 333),

Hive Cluster Errors

Vous pouvez généralement trouver la cause d'une erreur Hive dans le fichier `syslog`, dont le lien est disponible dans le volet Étapes. Si vous ne pouvez pas déterminer le problème grâce à ce fichier, vérifiez le message d'erreur de la tentative de tâche Hadoop. Vous y accédez grâce au lien disponible dans le volet Tentatives de tâche.

Les erreurs suivantes sont communes aux clusters Hive.

Rubriques

- Are you using the latest version of Hive? (p. 452)
- Did you encounter a syntax error in the Hive script? (p. 452)
- Did a job fail when running interactively? (p. 452)
- Are you having trouble loading data to or from Amazon S3 into Hive? (p. 452)

Are you using the latest version of Hive?

La dernière version de Hive comporte tous les correctifs et correctifs de bogues actuels, ce qui peut résoudre votre problème.

Did you encounter a syntax error in the Hive script?

Si une étape échoue, consultez le fichier `stdout` des journaux relatifs à l'étape dans laquelle le script Hive a été exécuté. Si l'erreur n'est pas indiquée dans ce fichier, consultez le fichier `syslog` des journaux de la tentative de tâche qui a échoué. Pour plus d'informations, consultez View Log Files (p. 333),

Did a job fail when running interactively?

Si vous exécutez Hive de façon interactive sur le nœud principal et si le cluster a échoué, consultez les entrées du journal `syslog` dans le journal des tentatives de tâche afin d'identifier la tentative de tâche qui a échoué. Pour plus d'informations, consultez View Log Files (p. 333),

Are you having trouble loading data to or from Amazon S3 into Hive?

Si vous rencontrez des difficultés pour accéder aux données dans Amazon S3, commencez par vérifier les causes possibles répertoriées dans Are you experiencing trouble loading data to or from Amazon

S3? (p. 441). Si aucun de ces problèmes n'est à l'origine, vous pouvez utiliser les options spécifiques à Hive suivantes.

- Make sure you are using the latest version of Hive, which has all the current patches and bug fixes that may resolve your issue. For more information, see Apache Hive.
- Using `INSERT OVERWRITE` requires listing the contents of the Amazon S3 bucket or folder. This is an expensive operation. If possible, manually prune the path instead of having Hive list and delete the existing objects.
- If you use Amazon EMR release versions earlier than 5.0, you can use the following command in HiveQL to pre-cache the results of an Amazon S3 list operation locally on the cluster:

```
set hive.optimize.s3.query=true;
```

- Use static partitions where possible.
- In some versions of Hive and Amazon EMR, it is possible that using ALTER TABLES will fail because the table is stored in a different location than expected by Hive. The solution is to add or update following in `/home/hadoop/conf/core-site.xml`:

```
<property>
    <name>fs.s3n.endpoint</name>
    <value>s3.amazonaws.com</value>
</property>
```

VPC Errors

Les erreurs suivantes sont communes à la configuration de VPC dans Amazon EMR.

Rubriques

- Invalid Subnet Configuration (p. 453)
- Missing DHCP Options Set (p. 453)
- Permissions Errors (p. 454)
- Errors That Result in ÉCHEC_DÉMARRAGE (p. 455)
- Cluster Terminé avec des erreurs and NameNode Fails to Start (p. 455)

Invalid Subnet Configuration

Sur la page Cluster Details (Détails de cluster), dans le champ Status (État), vous voyez une erreur similaire à ce qui suit :

```
The subnet configuration was invalid: Cannot find route to InternetGateway in
main RouteTable rtb-id for vpc vpc-id.
```

Pour résoudre ce problème, vous devez créer une passerelle Internet et la connecter à votre VPC. Pour plus d'informations, consultez Ajout d'une passerelle Internet à votre VPC.

Sinon, vérifiez que vous avez configuré votre VPC avec les paramètres Enable DNS resolution (Activer la résolution DNS) et Enable DNS hostname support (Activer le support de nom d'hôte DNS) activés. Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.

Missing DHCP Options Set

Vous voyez un échec d'étape dans le journal système (syslog) de cluster avec une erreur similaire à ce qui suit :

```
ERROR org.apache.hadoop.security.UserGroupInformation (main):
PriviledgedActionException as:hadoop (auth:SIMPLE) cause:java.io.IOException:
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

ou

```
ERROR org.apache.hadoop.streaming.StreamJob (main): Error Launching job :
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

Pour résoudre ce problème, vous devez configurer un VPC incluant un jeu d'options DHCP dont les paramètres sont définis sur les valeurs suivantes :

> **Note**
>
> Si vous utilisez la région AWS GovCloud (US-West), définissez le nom de domaine **us-gov-west-1.compute.internal** au lieu de la valeur utilisée dans l'exemple suivant.

- domain-name = **ec2.internal**

  Use **ec2.internal** if your region is USA Est (Virginie du Nord). For other regions, use *region-name*.**compute.internal**. For example in us-west-2, use domain-name=**us-west-2.compute.internal**.
- domain-name-servers = **AmazonProvidedDNS**

Pour plus d'informations, consultez Jeux d'options DHCP.

Permissions Errors

Une défaillance dans le journal `stderr` pour une étape indique qu'une ressource Amazon S3 n'a pas les autorisations appropriées. Il s'agit d'une erreur 403 et l'erreur ressemble à :

```
Exception in thread "main" com.amazonaws.services.s3.model.AmazonS3Exception: Access Denied
 (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: REQUEST_ID
```

Si ActionOnFailure est définie sur `TERMINATE_JOB_FLOW`, alors cela entraîne un arrêt du cluster avec l'état `SHUTDOWN_COMPLETED_WITH_ERRORS`.

Quelques façons pour résoudre ce problème incluent les suivantes :

- If you are using an Amazon S3 bucket policy within a VPC, make sure to give access to all buckets by creating a VPC endpoint and selecting Allow all under the Policy option when creating the endpoint.
- Make sure that any policies associated with S3 resources include the VPC in which you launch the cluster.
- Try running the following command from your cluster to verify you can access the bucket

  ```
  hadoop fs -copyToLocal s3://path-to-bucket /tmp/
  ```

- You can get more specific debugging information by setting the `log4j.logger.org.apache.http.wire` parameter to `DEBUG` in `/home/hadoop/conf/log4j.properties` file on the cluster. You can check the `stderr` log file after trying to access the bucket from the cluster. The log file will provide more detailed information:

  ```
  Access denied for getting the prefix for bucket - us-west-2.elasticmapreduce with path
   samples/wordcount/input/
  15/03/25 23:46:20 DEBUG http.wire: >> "GET /?prefix=samples%2Fwordcount%2Finput
  %2F&delimiter=%2F&max-keys=1 HTTP/1.1[\r][\n]"
  ```

```
15/03/25 23:46:20 DEBUG http.wire: >> "Host: us-
west-2.elasticmapreduce.s3.amazonaws.com[\r][\n]"
```

Errors That Result in `ÉCHEC_DÉMARRAGE`

Avant AMI 3.7.0, pour les vpc où un nom d'hôte est spécifié, Amazon EMR mappe les noms d'hôte internes du sous-réseau avec des adresses de domaine personnalisées comme suit : `ip-X.X.X.X.customdomain.com.tld`. Par exemple, si le nom d'hôte était `ip-10.0.0.10` et le VPC a l'option de nom de domaine définie sur customdomain.com, le nom d'hôte qui en résulte est mappé par Amazon EMR serait `ip-10.0.1.0.customdomain.com`. Une entrée est ajoutée dans `/etc/hosts` pour résoudre le nom d'hôte à 10.0.0.10. Ce comportement est modifié avec l'AMI 3.7.0 et Amazon EMR respecte désormais entièrement la configuration DHCP du VPC. Précédemment, les clients pouvaient également utiliser une action d'amorçage pour spécifier un mappage de nom d'hôte.

Si vous souhaitez conserver ce comportement, vous devez fournir le DNS et transmettre la configuration de résolution dont vous avez besoin pour le domaine personnalisé.

Cluster `Terminé avec des erreurs` and NameNode Fails to Start

Lors du lancement d'un cluster EMR dans un VPC qui permet d'utiliser un nom de domaine DNS personnalisé, votre cluster peut échouer avec le message d'erreur suivant dans la console :

```
Terminated with errors  On the master instance(instance-id), bootstrap action 1 returned a
 non-zero return code
```

L'échec est un résultat de l'incapacité de NameNode à démarrer. Cela entraînera l'erreur suivante trouvée dans les journaux de l'enregistreur de noms, dont Amazon S3 URI est du formulaire : `s3://mybucket/logs/cluster-id/daemons/master instance-id/hadoop-hadoop-namenode-master node hostname.log.gz`:

```
2015-07-23 20:17:06,266 WARN
      org.apache.hadoop.hdfs.server.namenode.FSNamesystem (main): Encountered  exception
      loading fsimage  java.io.IOException: NameNode is not formatted.
      at

 org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:212)
           at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:1020)
           at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:739)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:537)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:596)
      at  org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:765)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:749)
      at
      org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1441)
           at
      org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1507)
```

Cela est dû à un problème potentiel où une instance EC2 peut avoir plusieurs jeux de noms de domaine pleinement qualifiés lors du lancement de clusters EMR dans un VPC, utilisant à la fois un serveur DNS

fourni par AWS et un serveur DNS personnalisé fourni par l'utilisateur. Si le serveur DNS fourni par l'utilisateur ne fournit aucun enregistrement de pointeur (PTR) pour aucun enregistrement A utilisé pour désigner des nœuds dans un cluster EMR, les clusters échouent à démarrer lorsqu'ils sont configurés de cette façon. La solution consiste à ajouter 1 enregistrement PTR pour chaque enregistrement A créé lorsqu'une instance EC2 est démarrée dans un des sous-réseaux dans le VPC.

AWS GovCloud (US-West) Errors

La région AWS GovCloud (US-West) diffère des autres régions dans sa sécurité, sa configuration et ses paramètres par défaut. Par conséquent, utilisez la liste de contrôle suivante pour corriger des erreurs Amazon EMR qui sont spécifiques à la région AWS GovCloud (US-West) avant d'utiliser des recommandations plus générales de dépannage.

- Verify that your IAM roles are correctly configured. For more information, see Configure IAM Service Roles for Amazon EMR Permissions to AWS Services and Resources (p. 213).
- Ensure that your VPC configuration has correctly configured DNS resolution/hostname support, Internet Gateway, and DHCP Option Set parameters. For more information, see VPC Errors (p. 453).

Si ces étapes ne corrigent pas le problème, poursuivez avec les étapes de dépannage des erreurs Amazon EMR courantes. Pour plus d'informations, consultez Common Errors in Amazon EMR (p. 440),

Other Issues

Do you not see the cluster you expect in the Cluster List page or in results returned from ListClusters API?

Vérifiez les éléments suivants :

- The cluster age is less than two months. Amazon EMR preserves metadata information about completed clusters for your reference, at no charge, for two months. The console does not provide a way to delete completed clusters from the console; these are automatically removed for you after two months.
- You have permissions to view the cluster.
- You are viewing the correct region.

Dépannage d’un cluster Lake Formation (version bêta)

Cette section vous guide à travers le processus de dépannage des problèmes courants lors de l'utilisation d'Amazon EMR avec AWS Lake Formation.

Expiration de session

Le délai d'expiration de session pour les blocs-notes EMR et Zeppelin est contrôlé par le paramètre `Maximum CLI/API session duration` du rôle IAM pour Lake Formation. La valeur par défaut de ce paramètre est une heure. Lorsqu'un délai d'expiration de session se produit, le message suivant s’affiche dans la sortie de vos entrées de bloc-notes lorsque vous essayez d'exécuter des commandes Spark SQL.

```
Error 401    HTTP ERROR: 401 Problem accessing /sessions/2/statements.
Reason:  JWT token included in request failed validation.
```

```
Powered by Jetty:// 9.3.24.v20180605
 org.springframework.web.client.HttpClientErrorException: 401 JWT token included in request
 failed validation…
```

Pour valider votre session, actualisez la page. Vous serez invité à vous authentifier à nouveau à l'aide de votre IdP et serez redirigé vers le bloc-notes. Vous pouvez continuer à exécuter des requêtes après vous être authentifié à nouveau.

Aucune autorisation pour l'utilisateur sur la table demandée

Si vous essayez d'accéder à une table à laquelle vous n'avez pas accès, l'exception suivante apparaîtra dans la sortie de vos entrées de bloc-notes lorsque vous tenterez d'exécuter des commandes SQL Spark.

```
org.apache.spark.sql.AnalysisException: org.apache.hadoop.hive.ql.metadata.HiveException:
 Unable to fetch table table.
Resource does not exist or requester is not authorized to access requested permissions.
(Service: AWSGlue; Status Code: 400; Error Code: AccessDeniedException; Request ID: …
```

Pour accéder à la table, vous devez accorder l'accès à l'utilisateur en mettant à jour les autorisations associées à cette table dans Lake Formation.

Insertion, création et modification de tables : non prises en charge en version bêta

L'insertion, la création ou la modification de tables dans des bases de données protégées par des stratégies Lake Formation ne sont pas prises en charge. Si vous effectuez ces opérations, l'exception suivante apparaîtra dans la sortie de vos entrées de bloc-notes lorsque vous tenterez d'exécuter des commandes Spark SQL :

```
java.io.IOException:
 com.amazon.ws.emr.hadoop.fs.shaded.com.amazonaws.services.s3.model.AmazonS3Exception:
            Access Denied (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied;
 Request ID: …
```

Pour plus d'informations, consultez Limitations de l'intégration d'Amazon EMR à AWS Lake Formation.

Write Applications that Launch and Manage Clusters

Rubriques

- End-to-End Amazon EMR Java Source Code Sample (p. 458)
- Common Concepts for API Calls (p. 461)
- Use SDKs to Call Amazon EMR APIs (p. 462)

Vous pouvez accéder aux fonctionnalités fournies par l'API Amazon EMR en appelant des fonctions d'habillage dans l'un des kit SDK AWS. Les kits AWS SDK fournissent des fonctions propres au langage qui enveloppent l'API du service web et simplifient la connexion au service web, gérant un grand nombre des détails de connexion pour vous. Pour plus d'informations sur l'appel d'Amazon EMR à l'aide d'un des kits SDK, consultez Use SDKs to Call Amazon EMR APIs (p. 462).

> **Important**
>
> Le taux de demandes maximal pour Amazon EMR est d'une demande toutes les dix secondes.

End-to-End Amazon EMR Java Source Code Sample

Les développeurs peuvent appeler l'API Amazon EMR à l'aide d'un code Java personnalisé pour faire les mêmes choses avec la console ou l'interface de ligne de commande Amazon EMR. Cette section fournit les étapes de bout en bout nécessaires pour installer AWS Toolkit for Eclipse et exécuter un exemple de code source Java pleinement fonctionnel qui ajoute des étapes à un cluster Amazon EMR.

> **Note**
>
> Cet exemple se concentre sur Java, mais Amazon EMR prend également en charge plusieurs langages de programmation avec un ensemble de kits SDK Amazon EMR. Pour plus d'informations, consultez Use SDKs to Call Amazon EMR APIs (p. 462),

Cet exemple de code source Java montre comment effectuer les tâches suivantes à l'aide de l'API Amazon EMR :

- Retrieve AWS credentials and send them to Amazon EMR to make API calls
- Configure a new custom step and a new predefined step
- Add new steps to an existing Amazon EMR cluster
- Retrieve cluster step IDs from a running cluster

> **Note**
>
> Cet exemple montre comment ajouter des étapes à un cluster existant et nécessite donc que vous ayez un cluster actif sur votre compte.

Avant de commencer, installez une version de l'Eclipse IDE for Java EE Developers (IDE Eclipse pour développeurs Java EE) qui correspond à votre plate-forme informatique. Pour plus d'informations, consultez Eclipse Downloads (Téléchargements Eclipse).

Ensuite, installez le plug-in de développement de base de données pour Eclipse.

Pour installer le plug-in Eclipse de développement de base de données

1. Ouvrez l'IDE Eclipse.
2. Choisissez Help (Aide) et Install New Software (Installer un nouveau logiciel).
3. Dans le champ Work with: (Travailler avec :), saisissez **`http://download.eclipse.org/releases/kepler`** ou le chemin d'accès qui correspond au numéro de version de votre IDE Eclipse.
4. Dans la liste des éléments, choisissez Database Development (Développement de base de données) et Finish (Terminer).
5. Redémarrez Eclipse lorsque vous y êtes invité.

Ensuite, installez le Toolkit for Eclipse pour rendre disponibles les modèles de projet de code source préconfigurés utiles.

Pour installer le Toolkit for Eclipse

1. Ouvrez l'IDE Eclipse.
2. Choisissez Help (Aide) et Install New Software (Installer un nouveau logiciel).
3. Dans le champ Travailler avec :, saisissez **`https://aws.amazon.com/eclipse`**.
4. Dans la liste des éléments, choisissez AWS Toolkit for Eclipse (Boîte à outils AWS pour Eclipse) et Finish (Terminer).
5. Redémarrez Eclipse lorsque vous y êtes invité.

Ensuite, créez un projet Java AWS et exécutez l'exemple de code source Java.

Pour créer un projet Java AWS

1. Ouvrez l'IDE Eclipse.
2. Choisissez File (Fichier), New (Nouveau) et Other (Autre).
3. Dans la boîte de dialogue Select a wizard (Sélectionner un assistant), choisissez AWS Java Project (Projet Java AWS) et Next (Suivant).
4. Dans la boîte de dialogue New AWS Java Project (Nouveau projet Java AWS), dans le champ **`Project name:`**, saisissez le nom de votre nouveau projet, par exemple **`EMR-sample-code`**.
5. Choisissez Configure AWS accounts… (Configuration des comptes AWS...), saisissez vos clés d'accès publiques et privées, puis choisissez Finish (Terminer). Pour plus d'informations sur la création des clés d'accès, voir Comment obtenir des informations d'identification de sécurité ? dans le Amazon Web Services General Reference.

   > **Note**
   >
   > Vous devez not incorporer les clés d'accès directement dans le code. Le kit SDK Amazon EMR vous permet de placer les clés d'accès à des emplacements connus afin que vous n'ayez pas besoin de les garder dans le code.

6. Dans le nouveau projet Java, cliquez avec le bouton droit sur le dossier src, puis choisissez New (Nouveau) et Class (Classe).
7. Dans la boîte de dialogue Java Class (Classe Java), dans le champ Name (Nom), saisissez un nom pour votre nouvelle classe, par exemple **`main`**.
8. Dans la section Which method stubs would you like to create? (Quels bouchons souhaitez-vous créer ?) choisissez public static void main(String[] args) et Finish (Terminer).
9. Entrez le code source Java à l'intérieur de votre nouvelle classe et ajoutez les instructions import appropriées pour les classes et les méthodes figurant dans cet exemple. Pour plus de commodité, le code source complet est présenté ci-dessous.

Note

Dans le code d'échantillon suivant, remplacez l'exemple d'ID du cluster (jobflottwid), *j-xxxxxxxxxxxx*, avec un ID de cluster valide dans votre compte trouvé dans le AWS Management Console ou en utilisant la commande CLI AWS suivante :

```
aws emr list-clusters --active | grep "Id"
```

De plus, remplacez l'exemple Amazon S3 chemin, *s3 ://path/to/my/jarfolder*, avec le chemin valide vers votre JAR. Enfin, remplacez le nom de la classe, *com.my.Main1*, avec le nom correct de la classe dans votre JAR, le cas échéant.

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

 public static void main(String[] args) {
  AWSCredentials credentials_profile = null;
  try {
   credentials_profile = new ProfileCredentialsProvider("default").getCredentials();
        } catch (Exception e) {
            throw new AmazonClientException(
                    "Cannot load credentials from .aws/credentials file. " +
                    "Make sure that the credentials file exists and the profile name is
 specified within it.",
                    e);
        }

  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentials_profile))
   .withRegion(Regions.US_WEST_1)
   .build();

  // Run a bash script using a predefined step in the StepFactory helper class
     StepFactory stepFactory = new StepFactory();
     StepConfig runBashScript = new StepConfig()
       .withName("Run a bash script")
       .withHadoopJarStep(stepFactory.newScriptRunnerStep("s3://jeffgoll/emr-scripts/
create_users.sh"))
       .withActionOnFailure("CONTINUE");

     // Run a custom jar file as a step
     HadoopJarStepConfig hadoopConfig1 = new HadoopJarStepConfig()
        .withJar("s3://path/to/my/jarfolder") // replace with the location of the jar
 to run as a step
        .withMainClass("com.my.Main1") // optional main class, this can be omitted if
 jar above has a manifest
        .withArgs("--verbose"); // optional list of arguments to pass to the jar
     StepConfig myCustomJarStep = new StepConfig("RunHadoopJar", hadoopConfig1);

     AddJobFlowStepsResult result = emr.addJobFlowSteps(new AddJobFlowStepsRequest()
    .withJobFlowId("j-xxxxxxxxxxxx") // replace with cluster id to run the steps
    .withSteps(runBashScript,myCustomJarStep));

          System.out.println(result.getStepIds());
```

```
 }
}
```

10. Choisissez Run (Exécuter), Run As (Exécuter sous) et Java Application (Application Java).
11. Si l'exemple s'exécute correctement, une liste d'ID pour les nouvelles étapes s'affiche dans la fenêtre de la console IDE Eclipse. La sortie correcte est similaire à ce qui suit :

```
[s-39BLQZRJB2E5E, s-1L6A4ZU2SAURC]
```

Common Concepts for API Calls

Rubriques

- Endpoints for Amazon EMR (p. 461)
- Specifying Cluster Parameters in Amazon EMR (p. 461)
- Availability Zones in Amazon EMR (p. 462)
- How to Use Additional Files and Libraries in Amazon EMR Clusters (p. 462)

Lorsque vous écrivez une application qui appelle l'API Amazon EMR, plusieurs concepts s'appliquent lors de l'appel de l'une des fonctions de wrapper d'un kit SDK.

Endpoints for Amazon EMR

Un point de terminaison est une URL qui est le point d'entrée d'un service web. Chaque demande de service web doit contenir un point de terminaison. Ce point de terminaison spécifie la région AWS où les clusters sont créés, décrits ou arrêtés. Il a le formulaire `elasticmapreduce.`*`regionname`*`.amazonaws.com`. Si vous spécifiez le critère d'évaluation général (`elasticmapreduce.amazonaws.com`), Amazon EMR dirige votre demande vers un point de terminaison dans la région par défaut. Pour les comptes créés le 8 mars 2013 ou après cette date, la région par défaut est us-west-2. Pour les anciens comptes, la région par défaut est us-east-1.

Pour plus d'informations sur les régions et les points de terminaison pour Amazon EMR, consultez Régions et Points de terminaison dans le Référence générale d'Amazon Web Services.

Specifying Cluster Parameters in Amazon EMR

Les paramètres `Instances` vous permettent de configurer le type et le nombre d'instances EC2 pour créer des nœuds afin de traiter les données. Hadoop répartit le traitement des données entre plusieurs nœuds du cluster. Le nœud maître est responsable du suivi de l'intégrité des nœuds principaux et de tâches, et de l'interrogation des nœuds pour obtenir le statut des résultats des travaux. Les nœuds principaux et de tâches effectuent le traitement réel des données. Si vous possédez un cluster à nœud unique, ce nœud fait office de nœud maître et principal.

Le paramètre `KeepJobAlive` d'une demande `RunJobFlow` détermine s'il convient d'arrêter le cluster lorsqu'il n'a plus d'étapes de cluster à exécuter. Définissez cette valeur sur `False` lorsque vous savez que le cluster s'exécute comme prévu. Lorsque vous dépannez le flux de travail et que vous ajoutez des étapes pendant que l'exécution du cluster est suspendue, définissez la valeur sur `True`. Cela réduit le temps et les dépenses de téléchargement des résultats vers Amazon Simple Storage Service (Amazon S3), uniquement pour répéter le processus après avoir modifié une étape pour redémarrer le cluster.

Si `KeepJobAlive` a pour valeur `true`, après avoir réussi à terminer le travail du cluster, vous devez envoyer une demande `TerminateJobFlows` ou le cluster continuera à s'exécuter et à générer des frais AWS.

Pour de plus amples informations sur les paramètres spécifiques à `RunJobFlow`, veuillez consulter RunJobFlow. Pour de plus amples informations sur les paramètres génériques dans la demande, veuillez consulter Paramètres de demande communs.

Availability Zones in Amazon EMR

Amazon EMR utilise des instances EC2 comme nœuds pour traiter les clusters. Ces instances EC2 ont des emplacements composés de régions et de zones de disponibilité. Les régions sont dispersées et situées dans des zones géographiques distinctes. Les zones de disponibilité sont des emplacements distincts dans une région, isolés des défaillances dans d'autres zones de disponibilité. Chaque zone de disponibilité fournit une connectivité réseau économique à faible latence aux autres zones de disponibilité de la même région. Pour obtenir une liste des régions et points de terminaison pour Amazon EMR, consultez le chapitre Régions et points de terminaison dans le Référence générale d'Amazon Web Services.

Le paramètre `AvailabilityZone` spécifie l'emplacement général du cluster. Ce paramètre est facultatif et, en général, nous déconseillons son utilisation. Quand le paramètre `AvailabilityZone` n'est pas spécifié, Amazon EMR sélectionne automatiquement la meilleure valeur `AvailabilityZone` pour le cluster. Vous pouvez trouver ce paramètre utile si vous souhaitez placer vos instances avec d'autres instances existantes en cours d'exécution, et que votre cluster doit lire ou écrire des données à partir de ces instances. Pour de plus amples informations, veuillez consulter Amazon EC2 Guide de l'utilisateur pour les instances Linux.

How to Use Additional Files and Libraries in Amazon EMR Clusters

Parfois, vous pouvez apprécier d'utiliser des fichiers supplémentaires ou des bibliothèques personnalisées avec vos applications de mappage et de réduction. Par exemple, vous pouvez apprécier d'utiliser une bibliothèque qui convertit un fichier PDF en texte clair.

Pour mettre en cache un fichier afin que le mappeur ou le réducteur l'utilise dans le cadre du streaming Hadoop

- Dans le champ `args` du fichier JAR, ajoutez l'argument suivant :

  ```
  -cacheFile s3://bucket/path_to_executable#local_path
  ```

  Le fichier, `local_path`, se trouve dans le répertoire de travail du mappeur et peut faire référence au fichier.

Use SDKs to Call Amazon EMR APIs

Rubriques

- Using the AWS SDK for Java to Create an Amazon EMR Cluster (p. 463)

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 Kits SDK dans Outils pour Amazon Web Services.

Using the AWS SDK for Java to Create an Amazon EMR Cluster

Le kit AWS SDK pour Java fournit trois packages avec les fonctionnalité Amazon EMR :

- com.amazonaws.services.elasticmapreduce
- com.amazonaws.services.elasticmapreduce.model
- com.amazonaws.services.elasticmapreduce.util

Pour plus d'informations sur ces packages, consultez le AWS SDK for Java API Reference.

L'exemple suivant illustre la façon dont les kits SDK peuvent simplifier la programmation avec Amazon EMR. L'exemple de code ci-dessous utilise l'objet `StepFactory`, une classe d'annotation permettant de créer des types d'étapes Amazon EMR courants, pour créer un cluster Hive interactif avec la fonction débogage activée.

> **Note**
>
> Les identifiants statiques utilisés dans cet exemple peuvent expirer. N'utilisez pas un fournisseur d'informations d'identification statique pour gérer vos clusters EMR si les informations d'identification doivent être actualisées automatiquement.

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

 public static void main(String[] args) {
  AWSCredentials credentials_profile = null;
  try {
   credentials_profile = new ProfileCredentialsProvider("default").getCredentials(); //
 specifies any named profile in .aws/credentials as the credentials provider
        } catch (Exception e) {
            throw new AmazonClientException(
                    "Cannot load credentials from .aws/credentials file. " +
                    "Make sure that the credentials file exists and that the profile name
 is defined within it.",
                    e);
        }

  // create an EMR client using the credentials and region specified in order to create the
 cluster
  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentials_profile))
   .withRegion(Regions.US_WEST_1)
   .build();

        // create a step to enable debugging in the AWS Management Console
  StepFactory stepFactory = new StepFactory();
  StepConfig enabledebugging = new StepConfig()
      .withName("Enable debugging")
      .withActionOnFailure("TERMINATE_JOB_FLOW")
      .withHadoopJarStep(stepFactory.newEnableDebuggingStep());
```

```
        // specify applications to be installed and configured when EMR creates the cluster
  Application hive = new Application().withName("Hive");
  Application spark = new Application().withName("Spark");
  Application ganglia = new Application().withName("Ganglia");
  Application zeppelin = new Application().withName("Zeppelin");

  // create the cluster
  RunJobFlowRequest request = new RunJobFlowRequest()
          .withName("MyClusterCreatedFromJava")
          .withReleaseLabel("emr-5.20.0") // specifies the EMR release version label, we
 recommend the latest release
          .withSteps(enabledebugging)
          .withApplications(hive,spark,ganglia,zeppelin)
          .withLogUri("s3://path/to/my/emr/logs") // a URI in S3 for log files is required
 when debugging is enabled
          .withServiceRole("EMR_DefaultRole") // replace the default with a custom IAM
 service role if one is used
          .withJobFlowRole("EMR_EC2_DefaultRole") // replace the default with a custom EMR
 role for the EC2 instance profile if one is used
          .withInstances(new JobFlowInstancesConfig()
              .withEc2SubnetId("subnet-12ab34c56")
              .withEc2KeyName("myEc2Key")
              .withInstanceCount(3)
              .withKeepJobFlowAliveWhenNoSteps(true)
              .withMasterInstanceType("m4.large")
              .withSlaveInstanceType("m4.large"));

    RunJobFlowResult result = emr.runJobFlow(request);
    System.out.println("The cluster ID is " + result.toString());

 }

}
```

Au minimum, vous devez transmettre un rôle de service et le rôle jobflow correspondant à EMR_DefaultRole et EMR_EC2_DefaultRole, respectivement. Vous pouvez le faire en appelant cette commande d'interface de ligne de commande AWS pour le même compte. Tout d'abord, vérifiez si les rôles existent déjà :

```
aws iam list-roles | grep EMR
```

Le profil d'instance (EMR_EC2_DefaultRole) et le rôle de service (EMR_DefaultRole) sont affichés s'ils existent :

```
"RoleName": "EMR_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_DefaultRole"
            "RoleName": "EMR_EC2_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_EC2_DefaultRole"
```

Si les rôles par défaut n'existent pas, vous pouvez utiliser la commande d'interface de ligne de commande AWS suivante pour les créer :

```
aws emr create-default-roles
```

Glossaire AWS

Pour la terminologie AWS la plus récente, consultez le Glossaire AWS dans le document AWS General Reference.

Si nous fournissons une traduction de la version anglaise du guide, la version anglaise du guide aura préséance en cas de contradiction. La traduction sera une traduction automatique.