aws

Management Guide

Amazon EMR

Amazon EMR: Management Guide

Table of Contents

Présentation d'Amazon EMR 1
Présentation 1
Présentation des clusters et des nœuds 2
Soumettre des tâches à un cluster 2
Traitement des données 3
Présentation du cycle de vie du cluster 4
Avantages 6
Économies sur les coûts 7
AWS intégration 7
Déploiement 8
Capacité de mise à l'échelle et flexibilité 8
Fiabilité 9
Sécurité 10
Surveillance 11
Interfaces de gestion 12
Architecture 13
Stockage 13
Gestion des ressources de cluster 14
Cadres de traitement de données 15
Applications et programmes 16
Configuration d'Amazon EMR 17
Inscrivez-vous pour un Compte AWS 17
Création d'un utilisateur doté d'un accès administratif 17
Créer une paire de clés Amazon EC2 pour SSH 19
Étapes suivantes 20
Didacticiel de démarrage 21
Présentation 21
Étape 1 : Planifier et configurer 22
Préparation du stockage pour Amazon EMR 22
Préparation d'une application avec des données d'entrée pour Amazon EMR 23
Lancement d'un cluster Amazon EMR 25
Étape 2 : Gérer 28
Soumission de travail à Amazon EMR 28
Affichage des résultats 32

Étape 3 : Nettoyer 37
Arrêt de votre cluster 37
Suppression des ressources S3 39
Étapes suivantes 40
Découvrez les applications de big data pour Amazon EMR 40
Planification du matériel, de la mise en réseau et de la sécurité du cluster 40
Gestion des clusters 40
Utilisation d'une interface différente 40
Consultation du blog technique EMR 41
Console Amazon EMR 42
Fonctionnalités de la console 42
Résumé des différences 43
Compatibilité des clusters dans la console 43
Création de clusters 43
Affichage et recherche de clusters 45
Afficher ou modifier les détails du cluster 46
Différences lorsque vous travaillez avec des configurations de sécurité 47
Amazon EMR Studio 49
Fonctions principales 49
Historique des fonctionnalités 50
Comment ça marche 51
Authentification et connexion utilisateur 52
Contrôle d'accès 56
Espaces de travail 57
Stockage pour ordinateurs portables 58
Considérations 58
Considérations 58
Problèmes connus 61
Limites fonctionnelles 62
Service Limits 63
Bonnes pratiques en matière de VPC et de sous-réseaux 63
Exigences du cluster 64
Configurer EMR Studio 66
Autorisations d'administrateur pour créer un EMR Studio 67
Configurer un Amazon EMR Studio 73
Gérer un Studio 141

Chiffrer les blocs-notes d'espace de travail 148
Contrôler le trafic réseau d'EMR Studio 151
Création de modèles de clusters 153
Accès et autorisations pour les référentiels Git 159
Optimiser les tâches Spark 163
Utiliser un EMR Studio 164
Les bases de Workspace 165
Collaboration dans Workspace 173
Exécuter un Workspace avec un rôle d'exécution 176
Exécuter les blocs-notes Workspace par programmation 181
Consultation des données à l'aide de SQL Explorer 182
Attacher un calcul à un espace de travail 183
Lier les référentiels Git 191
Intégration à Athena 194
CodeWhisperer intégration 196
Déboguer des applications et des tâches 198
Installer les noyaux et les bibliothèques 203
Commandes magic 204
Utiliser des blocs-notes multilingues avec des noyaux Spark 213
Blocs-notes EMR 216
Ordinateurs portables dans la console 217
À propos de la transition 217
Que devez-vous faire ? 218
Avantages des espaces de travail 218
Autorisations nécessaires 219
Considérations 220
Exigences en matière de cluster 221
Différences de capacités en fonction de la version du cluster 222
Limites pour les blocs-notes EMR connectés simultanément 223
Versions de bloc-notes Jupyter et de Python 224
Considérations relatives à la sécurité 224
Création d'un bloc-notes 224
Utilisation des blocs-notes EMR 228
Compréhension de l'état du bloc-notes 228
Utilisation de l'éditeur de bloc-notes 230
Modification des clusters 231

Suppression des blocs-notes et des fichiers de bloc-notes 232
Partage de fichiers de bloc-notes 233
Exécution par programmation 234
Présentation 234
Autorisations 235
Limites 236
Exemples 236
Exemples de commande de l'interface CLI 237
Exemple de script du kit SDK Boto3 243
Exemple de script Ruby 246
Emprunt d'identité pour Spark 248
Configuration de l'emprunt d'identité d'un utilisateur Spark 248
Utilisation du widget de surveillance de tâche Spark 249
Sécurité 250
Installation et utilisation des noyaux et des bibliothèques 251
..... 252
Installation des noyaux et des bibliothèques Python sur le nœud primaire d'un cluster 252
Considérations et limites relatives aux bibliothèques adaptées aux blocs-notes 255
Travail avec des bibliothèques adaptées aux blocs-notes 256
– Association de référentiels Git à des blocs-notes EMR 257
Prérequis et considérations 258
Ajout d'un référentiel Git à Amazon EMR 262
Mise à jour ou suppression d'un référentiel Git 265
Association ou dissociation d'un référentiel Git 266
Création d'un nouveau bloc-notes avec un référentiel Git associé 268
Utilisation de référentiels Git dans un bloc-notes 269
Planification et configuration des clusters 271
Lancement rapide d'un cluster 271
Configuration de l'emplacement de cluster et du stockage de données 272
Choisissez une AWS région 273
Gestion du stockage et des systèmes de fichiers 274
Préparation des données d'entrée 279
Configuration d'un emplacement de sortie 300
Planification et configuration des nœuds primaires 307
Applications et fonctionnalités prises en charge 308
Lancer un cluster Amazon EMR doté de plusieurs nœuds primaires 318

Intégration d'Amazon EMR aux groupes de placement EC2 324
Considérations et bonnes pratiques 332
Clusters EMR sur AWS Outposts 335
Prérequis 335
Limites 335
Considérations relatives à la connectivité réseau 336
Création d'un cluster Amazon EMR sur AWS Outposts 337
Clusters EMR sur les Zones Locales AWS 339
Types d’instance pris en charge 339
Création d'un cluster Amazon EMR sur les Zones Locales 340
Configuration de Docker 341
Registres Docker 342
Configuration des registres Docker 343
Configuration de YARN pour accéder à Amazon ECR sur EMR 6.0.0 et versions antérieures 344
Contrôle de la mise hors service d'un cluster 346
Configuration d'un cluster pour qu'il continue ou se résilie après l'exécution de l'étape 347
Utilisation d'une politique de résiliation automatique 350
Utilisation de la protection contre la résiliation 358
Remplacement de nœuds malsains 364
Paramètres de protection par défaut pour le remplacement et la terminaison des nœuds 365
Configuration du remplacement de nœuds défectueux lorsque vous lancez un cluster 365
Configuration du remplacement de nœuds défectueux dans un cluster en cours d'exécution 367
Utilisation des AMI 368
Présentation 368
À l'aide de l'AMI par défaut 369
Utilisation d'une image AMI personnalisée 452
Modification de la version d’AL 465
Personnalisation du volume racine EBS 466
Configuration des logiciels de cluster 470
Création d'actions d'amorçage 471
Configuration du matériel et de la mise en réseau d'un cluster 477
Comprendre les types de nœuds 478
Configuration des instances Amazon EC2 481
Configuration de la journalisation et du débogage du cluster 1302

Fichiers journaux par défaut ... 1303
Archiver les fichiers journaux sur Amazon S3 ... 1304
Emplacements des journaux ... 1309
Activation de l'outil de débogage ... 1311
Informations relatives aux options de débogage ... 1313
Clusters de balise ... 1313
Restrictions liées aux étiquettes ... 1315
Balisage des ressources pour la facturation ... 1316
Ajout de balises à un cluster ... 1316
Affichage des balises sur un cluster ... 1320
Retrait des balises d'un cluster ... 1321
Intégration de pilotes et d'applications tierces ... 1323
Utilisation d'outils de business intelligence avec Amazon EMR ... 1323
Sécurité ... 1324
Sécurité du réseau et de l'infrastructure ... 1324
Mises à jour de l'AMI Amazon Linux par défaut ... 1325
AWS Identity and Access Management avec Amazon EMR ... 1326
Clusters à locataire unique et à locataires multiples ... 1327
Protection des données ... 1328
Contrôle d'accès aux données ... 1328
Configurations de la sécurité ... 1329
Création d'une configuration de sécurité ... 1330
Spécification d’une configuration de sécurité ... 1361
Protection des données ... 1363
Chiffrer les données au repos et en transit ... 1364
IAM avec Amazon EMR ... 1379
Public ciblé ... 1379
Authentification par des identités ... 1380
Gestion des accès à l’aide de politiques ... 1384
Fonctionnement d'Amazon EMR avec IAM ... 1387
Rôles d'exécution pour les étapes Amazon EMR ... 1395
Configuration des rôles de service pour Amazon EMR ... 1404
Exemples de politiques basées sur l’identité ... 1467
S3 Access Grants avec Amazon EMR ... 1508
Présentation ... 1508
Comment ça marche ... 1509

Considérations 1510
Lancez un cluster. 1511
`Lake Formation` 1512
`fallbackToIAM` 1513
Authentifier pour les nœuds de cluster 1514
Utilisation d'une paire de clés Amazon EC2 pour les informations d'identification SSH 1514
Utilisation de l'authentification Kerberos 1515
Utilisation de l'authentification LDAP 1554
Intégration d’Amazon EMR à Identity Center 1566
Présentation 1566
Fonctionnalités 1567
Premiers pas 1567
Considérations 1574
Intégration d'Amazon EMR avec Lake Formation 1576
Comment Amazon EMR fonctionne avec Lake Formation 1576
Prérequis 1577
Activation de Lake Formation avec Amazon EMR 1578
Hudi et Lake Formation 1583
Iceberg et Lake Formation 1585
Delta Lake et Lake Formation 1586
Considérations 1588
Intégration d'Amazon EMR avec Apache Ranger 1589
Présentation de Ranger 1590
Prise en charge des applications et limitations 1592
Configuration d'Amazon EMR pour Apache Ranger 1595
Plug-ins Apache Ranger 1613
Résolution des problèmes liés à Apache 1640
Utilisation des vues du catalogue de données AWS Glue (aperçu) 1644
Création d’un affichage du Catalogue de données 1645
Activation de l'accès à une vue du catalogue de données 1647
Interrogation d’un affichage du Catalogue de données 1649
Limites 1649
Contrôle du trafic réseau avec des groupes de sécurité 1650
Utilisation de groupes de sécurité gérés par Amazon EMR 1652
Utilisation des groupes de sécurité supplémentaires 1663
Spécification des groupes de sécurité 1664

Groupes de sécurité pour Blocs-notes EMR 1668
Blocage de l'accès public 1670
Validation de conformité 1676
Résilience 1677
Sécurité de l’infrastructure 1678
Connexion à Amazon EMR à l'aide d'un point de terminaison d'un VPC d'interface 1678
Gestion des clusters 1684
Connexion à un cluster 1684
Avant de vous connecter 1685
Connexion au nœud primaire à l'aide de SSH 1688
Soumission de travail à un cluster 1715
Ajouter des étapes à l'aide de la console 1716
Ajouter des étapes à l'aide de l'interface CLI 1721
Exécution de plusieurs étapes 1723
Affichage des étapes 1724
Annulation d'étapes 1725
Affichage et surveillance d'un cluster 1727
Afficher l'état et les détails d'un cluster 1727
Amélioration du débogage des étapes 1735
Afficher l'historique de l'application 1738
Afficher les fichiers journaux 1748
Afficher les instances de cluster dans Amazon EC2 1753
CloudWatch événements et indicateurs 1755
Affichage des métriques d'application d'un cluster avec Ganglia 1829
Enregistrement des appels d'API Amazon EMR AWS CloudTrail 1829
Utiliser la mise à l'échelle des clusters 1832
Considérations 1834
Mise à l'échelle gérée 1834
Mise à l'échelle automatique avec une politique personnalisée 1864
Redimensionner un cluster en cours d'exécution 1877
Expiration des délais de mise en service 1886
Réduction de capacité des clusters 1891
Arrêter un cluster 1895
Résilier depuis la console 1896
Résilier depuis la CLI 1898
Résilier depuis l'API 1899

Cloner un cluster 1899
Automatisation de clusters récurrents avec AWS Data Pipeline 1901
Résolution des problèmes liés à un cluster 1903
Outils de dépannage 1903
Consulter les détails du cluster 1904
Afficher les détails de l'erreur 1904
Exécuter des scripts et configurer des processus 1905
Afficher les fichiers journaux 1905
Surveiller les performances du cluster 1906
Afficher et redémarrer les processus 1906
Affichage des processus en cours d'exécution 1907
Arrêt et redémarrage des processus 1908
Erreurs courantes 1911
Codes d’erreur 1912
Erreurs de ressource 1926
Erreurs d'entrée et sortie 1938
Erreurs d'autorisations 1941
Erreurs de cluster Hive 1942
Erreurs VPC 1944
Erreurs de cluster de diffusion en continu 1948
Erreurs de cluster des fichiers JAR personnalisés 1950
AWS GovCloud Erreurs (ouest des États-Unis) 1950
Trouver un cluster manquant 1951
Dépannage des clusters ayant échoué 1951
Étape 1 : Rassembler des données sur le problème 1952
Étape 2 : Vérifier l'environnement 1952
Étape 3 : Examiner le dernier changement d'état 1954
Étape 4 : Examiner les fichiers journaux 1954
Étape 5 : Test du cluster étape par étape 1956
Résolution des problèmes de rapidité des clusters 1957
Étape 1 : Rassembler des données sur le problème 1957
Étape 2 : Vérifier l'environnement 1958
Étape 3 : Examiner les fichiers journaux 1960
Étape 4 : Vérifier l'état du cluster et de l'instance 1961
Étape 5 : Vérifiez les groupes suspendus 1963
Étape 6 : Passer en revue les paramètres de configuration 1964

Étape 7 : Examiner les données d'entrée 1967
Résoudre les problèmes liés à un cluster de Lake Formation 1967
L'accès au lac de données n'est pas autorisé 1967
Expiration de session 1968
Aucune autorisation pour l'utilisateur sur le tableau demandé 1968
Interrogation de données entre comptes partagées avec Lake Formation 1969
Insertion, création et modification de tableaux 1970
Écriture d'applications pour lancer et gérer des clusters 1971
Exemple de code source Java nd-to-end Amazon EMR 1971
Concepts communs pour les appels d'API 1976
Points de terminaison pour Amazon EMR 1976
Spécification de paramètres de cluster dans Amazon EMR 1976
Zones de disponibilité dans Amazon EMR 1977
Comment utiliser des fichiers et des bibliothèques supplémentaires dans les clusters Amazon EMR 1977
Utiliser les SDK pour appeler les API Amazon EMR 1978
Utilisation du AWS SDK for Java pour créer un cluster Amazon EMR 1978
Gérer les Service Quotas Amazon EMR 1981
Que sont les Service Quotas Amazon EMR 1981
Comment gérer les Service Quotas Amazon EMR 1982
Quand configurer des événements EMR dans CloudWatch 1982
Glossaire AWS 1986
..... mcmlxxxvii

Présentation d'Amazon EMR

Amazon EMR (anciennement Amazon Elastic MapReduce) est une plateforme de clusters gérés qui simplifie l'exécution de frameworks de mégadonnées, tels qu'Apache Hadoop et Apache Spark, AWS pour traiter et analyser de grandes quantités de données. Grâce à ces infrastructures et des projets open source connexes, vous pouvez traiter des données à des fins d'analyse et pour des charges de travail business intelligence. Amazon EMR vous permet également de transformer et de déplacer de grandes quantités de données vers et à partir d'autres bases et entrepôts de données AWS , comme 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 d'Amazon EMR.
- Didacticiel : Les premiers pas avec Amazon EMR : ce didacticiel vous permet de commencer à utiliser Amazon EMR rapidement.

Dans cette section :

- Présentation d'Amazon EMR
- Avantages offerts par l'utilisation d'Amazon EMR
- Présentation de l'architecture d'Amazon EMR

Présentation d'Amazon EMR

Cette rubrique fournit une présentation des clusters Amazon EMR, y compris de la façon de soumettre des tâches 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
- Soumettre des tâches à un cluster
- Traitement des données
- Présentation du cycle de vie du cluster

Présentation des clusters et des nœuds

Le composant central d'Amazon EMR est le cluster. Un cluster est une collection 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 primaire : 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 primaire effectue le suivi du statut des tâches et surveille l'état du cluster. Chaque cluster a un nœud primaire ; il est possible de créer un cluster à nœud unique avec seulement le nœud primaire.
- Nœud principal : 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.

Soumettre des tâches à un cluster

Lorsque vous exécutez un cluster sur Amazon EMR, vous avez plusieurs options quant à la façon de spécifier les tâches qui doivent être effectuées.

- 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 AWS CLI les étapes de soumission, qui peuvent contenir une ou plusieurs tâches. Pour plus d’informations, consultez Soumission de travail à un cluster.
- Créez un cluster, connectez-vous au nœud primaire 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 Guide de version Amazon EMR.

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

Soumettre des tâches directement aux applications

Vous pouvez soumettre des tâches et interagir directement avec le logiciel qui est installé dans votre cluster Amazon EMR. Pour cela, vous vous connectez généralement au nœud primaire 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 à un cluster.

Exécuter des 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 jeu 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 jeu de données d'entrée à l'aide d'un programme Hive.
4. Écriture d'un jeu de données de sortie.

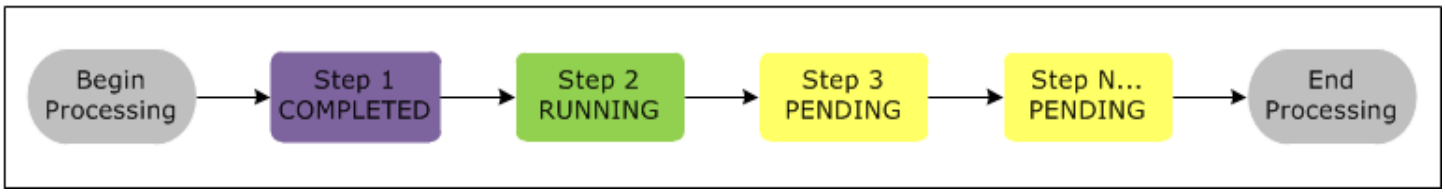
En règle générale, lorsque vous traitez des données dans Amazon EMR, l'entrée correspond à des données stockées sous forme de fichiers dans votre système de fichiers sous-jacent choisi, tel qu'Amazon S3 ou HDFS. Ces données passent d'une étape à l'autre dans la séquence de traitement. L'étape finale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

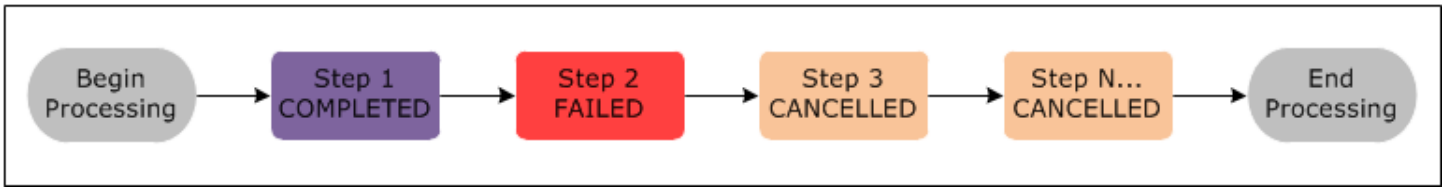
3. Lorsque la première étape de la séquence commence, son état passe à EN COURS D'EXÉCUTION. Les autres étapes restent à l'état EN SUSPENS.
4. Une fois la première étape terminée, son état devient TERMINÉ.
5. L'étape suivante de la séquence commence et son état passe à EN COURS D'EXÉCUTION. Une fois 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 FAILED. Vous pouvez déterminer ce qui se passe ensuite pour chaque étape. Par défaut, les étapes restantes de la séquence sont définies sur CANCELLED et ne sont pas exécutées si une étape précédente échoue. 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 :

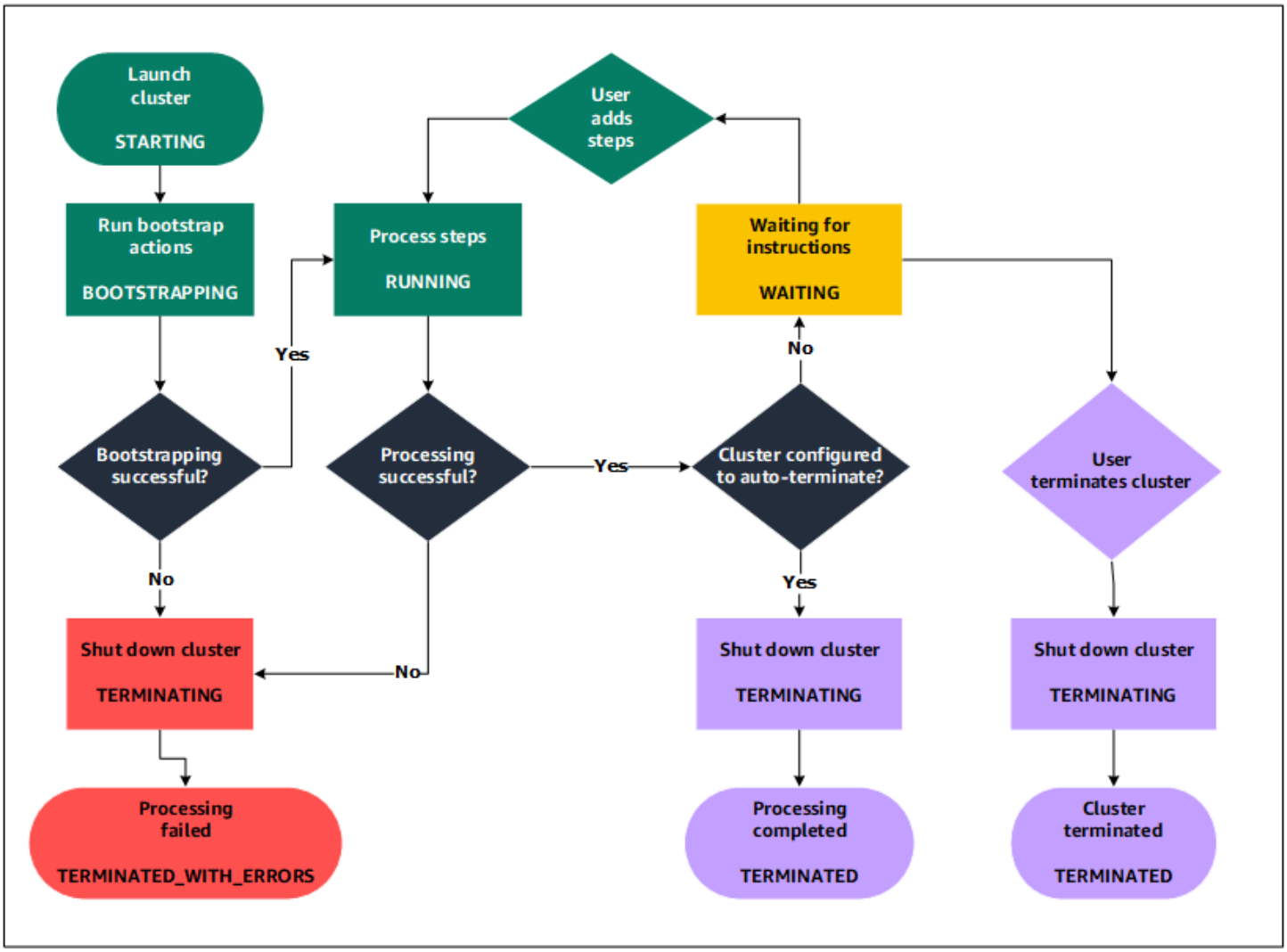
1. Amazon EMR alloue tout d'abord des instances EC2 dans le cluster pour chaque instance en fonction de vos spécifications. Pour plus d'informations, consultez Configuration du matériel et de la mise en réseau d'un cluster. Pour toutes les instances, Amazon EMR utilise l'AMI par défaut pour Amazon EMR ou une AMI Amazon Linux personnalisée que vous spécifiez. Pour plus

d'informations, consultez Utilisation d'une image AMI personnalisée. 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 Création d'actions d'amorçage pour installer des logiciels supplémentaires. 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 Soumission de travail à un cluster.
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 `TERMINATING` puis à l'état `TERMINATED`. Si le cluster est configuré pour attendre, vous devez l'arrêter manuellement lorsque vous n'en avez plus besoin. Après avoir arrêté manuellement le cluster, il passe à l'état `TERMINATING` puis à l'état `TERMINATED`.

En cas d'échec 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 `TERMINATED_WITH_ERRORS`. 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 Utilisation de la protection contre la résiliation.

Le schéma suivant représente le cycle de vie d'un cluster et la façon dont chaque étape du cycle de vie correspond à un état de cluster particulier.



Avantages offerts par 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

- Économies sur les coûts
- AWS intégration
- Déploiement
- Capacité de mise à l'échelle et flexibilité
- Fiabilité

- Sécurité
- Surveillance
- Interfaces de gestion

Économies sur les coûts

La tarification d'Amazon EMR dépend du type d'instance et du nombre d'instances Amazon 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 d'Amazon EMR.

> **Note**
>
> Lorsque vous configurez un cluster Amazon EMR dans un sous-réseau privé, nous vous recommandons de configurer également des points de terminaison VPC pour Amazon S3. Si votre cluster EMR se trouve dans un sous-réseau privé sans points de terminaison VPC pour Amazon S3, vous devrez payer des frais de passerelle NAT supplémentaires rattachés au trafic S3, car le trafic entre votre cluster EMR et S3 ne restera pas dans votre VPC.

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

AWS intégration

Amazon EMR s'intègre à d'autres AWS services afin de fournir des capacités et des fonctionnalités liées à la mise en réseau, au stockage, à la 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 devez lancer vos instances ;
- Amazon S3 pour stocker les données d'entrée et sortie ;
- Amazon va CloudWatch surveiller les performances du cluster et configurer les alarmes
- AWS Identity and Access Management (IAM) pour configurer les autorisations
- AWS CloudTrail pour auditer les demandes adressées au service
- AWS Data Pipeline pour planifier et démarrer vos clusters
- AWS Lake Formation pour découvrir, cataloguer et sécuriser les données dans un lac de données Amazon S3

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.

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 l'une des nombreuses distributions MapR. Amazon EMR utilise Amazon Linux. Vous pouvez donc également installer le logiciel sur votre cluster manuellement à l'aide du gestionnaire de packages yum ou à partir de la source. Pour plus d’informations, consultez Configuration des logiciels de cluster.

Capacité de mise à l'échelle et flexibilité

Amazon EMR offre une grande flexibilité pour augmenter ou réduire 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.

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 tâches. 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 ?.

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 primaires 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 informatiques en redimensionnant votre cluster et vous pouvez ajuster vos besoins de stockage en utilisant Amazon S3. Pour plus d'informations, consultez Gestion du stockage et des systèmes de fichiers.

Fiabilité

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

Amazon EMR fournit des options de configuration qui contrôlent la manière dont votre cluster est résilié, 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 Utilisation de la protection contre la résiliation.

Sécurité

Amazon EMR s'appuie sur d'autres AWS services, tels que IAM et Amazon VPC, ainsi que sur des fonctionnalités telles que les paires de clés Amazon EC2, pour 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 politiques IAM, que vous attachez à des utilisateurs ou à des groupes IAM. Les autorisations que vous définissez dans la politique déterminent les actions que les utilisateurs ou les membres du groupe peuvent effectuer et les ressources auxquelles ils peuvent accéder. Pour plus d'informations, consultez Fonctionnement d'Amazon EMR avec IAM.

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 autorisent le service et les instances à accéder à d'autres AWS services en votre nom. 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 des politiques AWS gérées, qui sont créées automatiquement pour vous la première fois que vous lancez un cluster EMR depuis 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 souhaitez plutôt gérer les autorisations AWS, vous pouvez choisir des rôles personnalisés pour le profil de service et d'instance. Pour plus d'informations, consultez Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..

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 primaire et un groupe de sécurité à partager par vos instances principales/de tâches. Amazon EMR configure les règles du groupe de sécurité afin de garantir la communication entre les instances du cluster. Si vous le souhaitez, vous pouvez configurer des groupes de sécurité supplémentaires et les affecter à vos instances primaires et principales/de tâches pour obtenir des règles plus avancées. Pour plus d'informations, consultez Contrôle du trafic réseau avec des groupes de sécurité.

Chiffrement

Amazon EMR prend en charge le chiffrement facultatif côté serveur et côté client d'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 racine pour le chiffrement côté client à l'aide du AWS Key Management Service (AWS KMS) ou de votre propre système de gestion des clés.

Pour plus d'informations, consultez Spécifier le chiffrement Amazon S3 à l'aide des propriétés EMRFS.

Amazon VPC

Amazon EMR prend en charge le lancement des clusters dans un cloud privé virtuel (VPC) dans Amazon VPC. Un VPC est un réseau virtuel isolé AWS qui permet de contrôler les aspects avancés de la configuration et de l'accès au réseau. Pour plus d’informations, consultez Configuration de la mise en réseau.

AWS CloudTrail

Amazon EMR s'intègre CloudTrail pour enregistrer les informations relatives aux demandes effectuées par ou au nom de votre AWS compte. 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 Enregistrement des appels d'API Amazon EMR AWS CloudTrail.

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 primaire. Vous pouvez utiliser le protocole réseau Secure Shell (SSH) 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 Utilisation d'une paire de clés Amazon EC2 pour les informations d'identification SSH.

Surveillance

Vous pouvez utiliser les interfaces de gestion et les fichiers journaux Amazon EMR pour résoudre les problèmes de cluster, tels que les échecs ou les erreurs. Amazon EMR permet d'archiver des fichiers

journaux dans Amazon S3 afin que vous puissiez stocker les journaux et résoudre les problèmes même après la résiliation de votre cluster. Amazon EMR fournit également un outil de débogage optionnel dans la console Amazon EMR pour parcourir les fichiers journaux en fonction des étapes, des travaux et des tâches. Pour plus d'informations, consultez Configuration de la journalisation et du débogage du cluster.

Amazon EMR s'intègre CloudWatch pour suivre les indicateurs de performance du cluster et les tâches 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 Amazon EMR avec CloudWatch.

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 l'adresse https://console.aws.amazon.com/elasticmapreduce/home.
- AWS Command Line Interface (AWS CLI) — Une application client que vous exécutez sur votre machine locale pour vous connecter à Amazon EMR et créer et gérer des clusters. AWS CLI Il contient un ensemble riche en fonctionnalités spécifiques 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 AWS CLI est la meilleure option. Pour plus d'informations et des exemples, consultez Amazon EMR dans la Référence des commandes AWS CLI .
- Kit de développement logiciel (SDK) : les kits SDK fournissent des fonctions pour appeler Amazon EMR afin de créer et de gérer des clusters. Ils vous permettent d'écrire des applications pour automatiser la création et la gestion des clusters. Le kit SDK est particulièrement recommandé si vous souhaitez étendre ou personnaliser les fonctionnalités 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 kits SDK, veuillez consulter les outils pour AWS et les 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 plus d'informations, consultez la Référence d'API Amazon EMR.

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
- Gestion des ressources de cluster
- Cadres de traitement de données
- Applications et programmes

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. Le HDFS est utile pour mettre en cache les résultats intermédiaires pendant le MapReduce traitement ou pour les charges de travail comportant des E/S aléatoires importantes.

Pour plus d'informations, consultez la rubrique Stockage d’instances de ce guide ou 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 proposés dans Amazon EMR qui n'utilisent pas YARN comme gestionnaire de ressources. Amazon EMR dispose également d'un agent sur chaque nœud, qui administre les composants YARN, maintient le cluster en bonne santé et communique avec le service Amazon EMR.

Les instances Spot étant souvent utilisées pour exécuter des nœuds de tâches, Amazon EMR dispose d'une fonctionnalité par défaut pour planifier les tâches YARN. Cela empêche l'échec de l'exécution des tâches lorsque les nœuds de tâches exécutés sur des instances Spot sont fermés. Pour ce faire, Amazon EMR autorise les processus principaux de l'application à s'exécuter uniquement sur les nœuds principaux. Le processus principal de l'application contrôle les tâches en cours d'exécution et doit rester actif pendant toute la durée de vie de la tâche.

Les versions 5.19.0 et ultérieures d'Amazon EMR utilisent la fonctionnalité intégrée d'étiquettes de nœuds YARN 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 afin que le planificateur de capacité YARN et le planificateur équitable tirent

parti des étiquettes des nœuds. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de manière à ce que les maîtres d'applications soient planifiés uniquement sur les nœuds portant le label CORE. La modification manuelle des propriétés rattachées dans les classifications de configuration de yarn-site et de capacity-scheduler, ou directement dans les fichiers XML rattachés, pourrait interrompre cette fonctionnalité ou la modifier.

Cadres 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 principaux frameworks de traitement disponibles pour Amazon EMR sont MapReduce Hadoop 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. Plusieurs frameworks sont disponibles MapReduce, tels que Hive, qui génère automatiquement les programmes Map et Reduce.

Pour plus d'informations, consultez la rubrique How map and reduce operations are actually carried out 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 big data. Comme Hadoop MapReduce, Spark est un système de traitement distribué open source qui utilise des graphes acycliques dirigés pour les plans d'exécution et une 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 la rubrique relative à Apache Spark sur les clusters Amazon EMR du Guide de version Amazon EMR.

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 de machine learning, 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 Guide de version Amazon EMR.

Configuration d'Amazon EMR

Exécutez les tâches décrites dans cette section avant de lancer un cluster Amazon EMR pour la première fois :

Avant d'utiliser Amazon EMR pour la première fois, exécutez les tâches suivantes :

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

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

Création d'un utilisateur doté d'un accès administratif

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

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

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

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

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

Connectez-vous en tant qu'utilisateur disposant d'un accès administratif

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

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

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

Pour obtenir des instructions, voir Création d'un ensemble d'autorisations dans le guide de AWS IAM Identity Center l'utilisateur.

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

   Pour obtenir des instructions, voir Ajouter des groupes dans le guide de AWS IAM Identity Center l'utilisateur.

Créer une paire de clés Amazon EC2 pour SSH

> **Note**
>
> Avec Amazon EMR versions 5.10.0 ou ultérieures, vous pouvez configurer Kerberos pour authentifier les utilisateurs et les connexions SSH sur un cluster. Pour plus d'informations, consultez Utilisation de Kerberos pour l'authentification avec Amazon EMR.

Pour vous authentifier et vous connecter aux nœuds d'un cluster via un canal sécurisé à l'aide du protocole Secure Shell (SSH), créez une paire de clés Amazon Elastic Compute Cloud (Amazon EC2) avant de lancer le 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.

Si...	Alors...
Vous possédez déjà une paire de clés Amazon EC2 que vous souhaitez utiliser, ou vous n'avez pas besoin de vous authentifier auprès de votre cluster.	Ignorez cette étape.
Vous avez besoin de créer une paire de clés.	Consultez la rubrique relative à la création de votre paire de clés à l'aide d'Amazon EC2.

Étapes suivantes

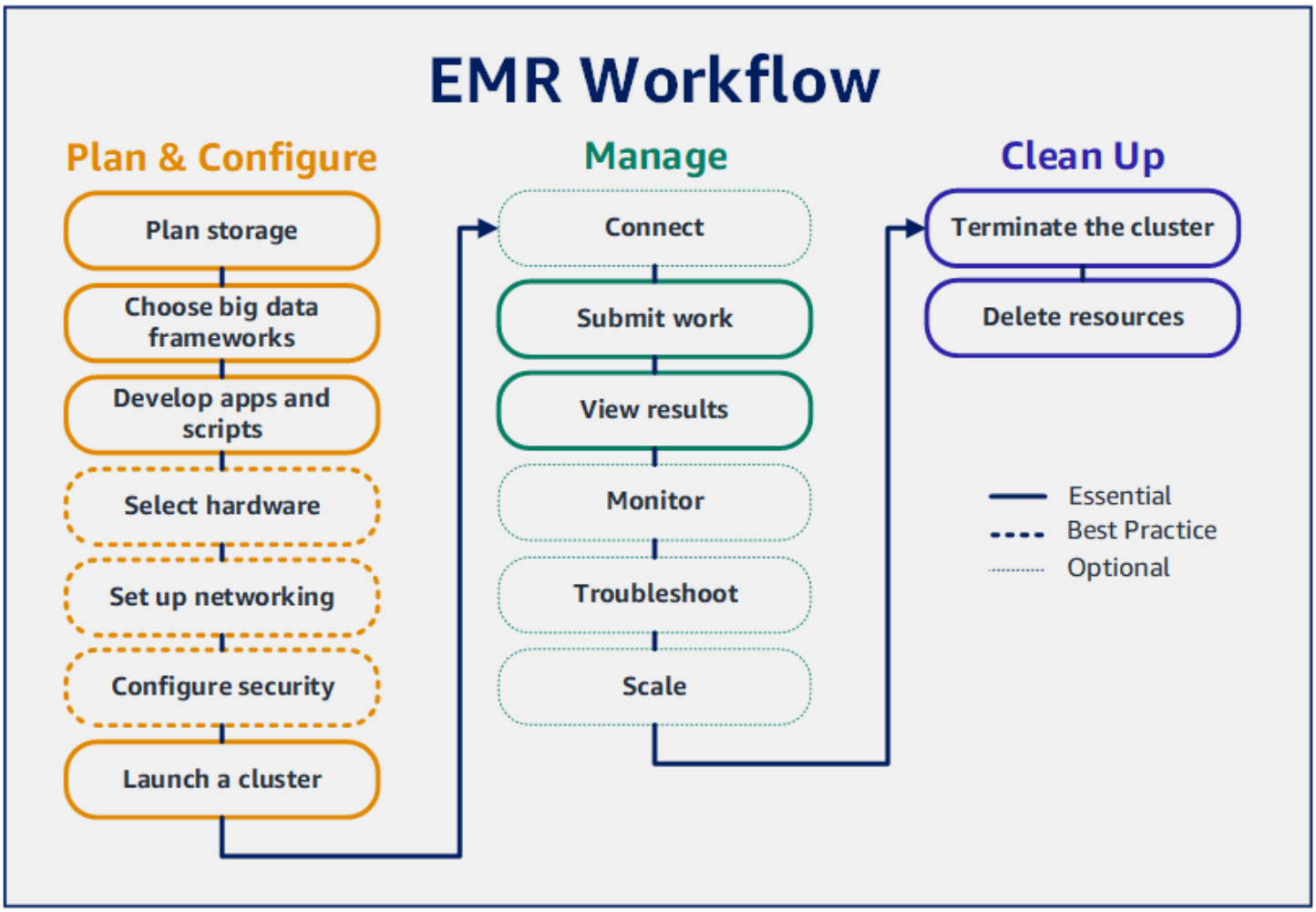
- Pour obtenir des conseils sur la création d'un exemple de cluster, consultez Didacticiel : Les premiers pas avec Amazon EMR.
- Pour plus d'informations sur la configuration d'un cluster personnalisé et le contrôle de l'accès à celui-ci, consultez Planification et configuration des clusters et Sécurité dans Amazon EMR.

Didacticiel : Les premiers pas avec Amazon EMR

Présentation

Avec Amazon EMR, vous pouvez configurer un cluster pour traiter et analyser des données avec des environnements de big data en quelques minutes seulement. Ce didacticiel explique comment lancer un exemple de cluster à l'aide de Spark et comment exécuter un PySpark script simple stocké dans un compartiment Amazon S3. Il couvre les tâches essentielles d'Amazon EMR dans trois catégories principales de flux de travail : planification et configuration, gestion et nettoyage.

Vous trouverez des liens vers des sujets plus détaillés au fur et à mesure que vous avancerez dans le didacticiel, ainsi que des idées d'étapes supplémentaires dans la section Étapes suivantes. Si vous avez des questions ou rencontrez des difficultés, contactez l'équipe d'Amazon EMR sur notre Forum de discussion.



Prérequis

- Avant de lancer un cluster Amazon EMR, assurez-vous d'avoir effectué les tâches décrites dans la rubrique Configuration d'Amazon EMR.

Coût

- L'exemple de cluster que vous créez s'exécute dans un environnement en direct. Le cluster enregistre des frais minimes. Pour éviter des frais supplémentaires, assurez-vous d'effectuer les tâches de nettoyage de la dernière étape de ce didacticiel. Les frais sont calculés au tarif par seconde conformément à la tarification Amazon EMR. Les frais varient également selon la région. Pour de plus amples informations, consultez la tarification Amazon EMR.
- Des frais minimes pourraient s'accumuler pour les petits fichiers que vous stockez dans Amazon S3. Certains ou tous les frais d'Amazon S3 peuvent être annulés si vous respectez les limites d'utilisation du niveau AWS gratuit. Pour plus d'informations, consultez la rubrique Tarification Amazon S3 et Niveau gratuit AWS.

Étape 1 : Planifier et configurer un cluster Amazon EMR

Préparation du stockage pour Amazon EMR

Lorsque vous utilisez Amazon EMR, vous pouvez choisir parmi différents systèmes de fichiers pour stocker les données d'entrée, les données de sortie et les fichiers journaux. Dans ce didacticiel, vous utilisez EMRFS pour stocker des données dans un compartiment S3. EMRFS est une implémentation du système de fichiers Hadoop qui permet de lire et d'écrire des fichiers normaux sur Amazon S3. Pour plus d'informations, consultez Gestion du stockage et des systèmes de fichiers.

Pour créer un compartiment pour ce didacticiel, suivez les instructions de la rubrique Comment créer un compartiment S3 ? dans le Guide de l'utilisateur de la console Amazon Simple Storage Service. Créez le compartiment dans la même AWS région où vous prévoyez de lancer votre cluster Amazon EMR. Par exemple, USA Ouest (Oregon) us-west-2.

Les compartiments et dossiers que vous utilisez avec Amazon EMR présentent les limites suivantes :

- Les noms peuvent être composés de lettres minuscules, de chiffres, de points (.) et de traits d'union (-).
- Les noms ne peuvent pas se terminer par des chiffres.

- Le nom d'un compartiment doit être unique pour tous les comptes AWS .
- Le dossier de sortie doit être vide.

Préparation d'une application avec des données d'entrée pour Amazon EMR

La façon la plus courante de préparer une application pour Amazon EMR consiste à télécharger l'application et ses données d'entrée sur Amazon S3. Ensuite, lorsque vous soumettez du travail à votre cluster, vous indiquez les emplacements Amazon S3 pour votre script et vos données.

Au cours de cette étape, vous chargez un exemple de PySpark script dans votre compartiment Amazon S3. Nous vous avons fourni un PySpark script que vous pouvez utiliser. Le script traite les données d'inspection des établissements alimentaires et renvoie un fichier de résultats dans votre compartiment S3. Le fichier des résultats répertorie les dix établissements ayant enregistré le plus grand nombre d'infractions de type « rouge ».

Vous chargez également des exemples de données d'entrée sur Amazon S3 pour que le PySpark script les traite. Les données d'entrée sont une version modifiée des résultats des inspections effectuées par le ministère de la Santé du comté de King, Washington, entre 2006 et 2020. Pour plus d'informations, consultez la rubrique Données ouvertes du comté de King : données sur l'inspection des établissements alimentaires. Voici des exemples de lignes du jeu de données.

```
name, inspection_result, inspection_closed_business, violation_type, violation_points
100 LB CLAM, Unsatisfactory, FALSE, BLUE, 5
100 PERCENT NUTRICION, Unsatisfactory, FALSE, BLUE, 5
7-ELEVEN #2361-39423A, Complete, FALSE, , 0
```

Pour préparer l'exemple de PySpark script pour EMR

1. Copiez l'exemple de code ci-dessous dans un nouveau fichier dans l'éditeur de votre choix.

```
import argparse

from pyspark.sql import SparkSession

def calculate_red_violations(data_source, output_uri):
    """
    Processes sample food establishment inspection data and queries the data to
 find the top 10 establishments
```

```
    with the most Red violations from 2006 to 2020.

    :param data_source: The URI of your food establishment data CSV, such as 's3://
DOC-EXAMPLE-BUCKET/food-establishment-data.csv'.
    :param output_uri: The URI where output is written, such as 's3://DOC-EXAMPLE-
BUCKET/restaurant_violation_results'.
    """
    with SparkSession.builder.appName("Calculate Red Health
 Violations").getOrCreate() as spark:
        # Load the restaurant violation CSV data
        if data_source is not None:
            restaurants_df = spark.read.option("header", "true").csv(data_source)

        # Create an in-memory DataFrame to query
        restaurants_df.createOrReplaceTempView("restaurant_violations")

        # Create a DataFrame of the top 10 restaurants with the most Red violations
        top_red_violation_restaurants = spark.sql("""SELECT name, count(*) AS
 total_red_violations
          FROM restaurant_violations
          WHERE violation_type = 'RED'
          GROUP BY name
          ORDER BY total_red_violations DESC LIMIT 10""")

        # Write the results to the specified output URI
        top_red_violation_restaurants.write.option("header",
 "true").mode("overwrite").csv(output_uri)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--data_source', help="The URI for you CSV restaurant data, like an S3
 bucket location.")
    parser.add_argument(
        '--output_uri', help="The URI where output is saved, like an S3 bucket
 location.")
    args = parser.parse_args()

    calculate_red_violations(args.data_source, args.output_uri)
```

2. Enregistrez le fichier sous le nom `health_violations.py`.

3. Chargez `health_violations.py` sur Amazon S3 dans le compartiment que vous avez créé pour ce didacticiel. Pour obtenir des instructions, consultez la rubrique Chargement d'un objet dans un compartiment dans le Guide de démarrage Amazon Simple Storage Service.

Préparation des exemples de données d'entrée pour EMR

1. Téléchargez le fichier zip food_establishment_data.zip.
2. Décompressez et enregistrez `food_establishment_data.zip` sous le nom `food_establishment_data.csv` sur votre machine.
3. Chargez le fichier CSV sur Amazon S3 dans le compartiment que vous avez créé pour ce didacticiel. Pour obtenir des instructions, consultez la rubrique Chargement d'un objet dans un compartiment dans le Guide de démarrage Amazon Simple Storage Service.

Pour plus d'informations sur la configuration des données pour EMR, consultez Préparation des données d'entrée.

Lancement d'un cluster Amazon EMR

Après avoir préparé un emplacement de stockage et votre application, vous pouvez lancer un exemple de cluster Amazon EMR. Au cours de cette étape, vous lancez un cluster Apache Spark à l'aide de la dernière version d'Amazon EMR.

Console

Pour lancer un cluster avec Spark installé avec la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Dans le volet de navigation, sous EMR on EC2, sélectionnez Clusters, puis Créer un cluster.
3. Sur la page Créer un cluster, notez les valeurs par défaut pour la version, le type d'instance, le nombre d'instances et les autorisations. Ces champs sont automatiquement renseignés avec des valeurs qui fonctionnent pour les clusters à usage général.
4. Dans le champ Nom du cluster, saisissez un nom de cluster unique pour vous aider à identifier votre cluster, tel que *`Mon premier cluster`*. Le nom de votre cluster ne peut pas contenir les caractères <, >, $, | ou `(backtick).
5. Sous Applications, choisissez l'option Spark pour installer Spark sur votre cluster.

> **Note**
>
> Choisissez les applications que vous souhaitez installer sur votre cluster Amazon EMR avant de lancer ce cluster. Il est impossible d'ajouter ou de supprimer des applications d'un cluster après son lancement.

6. Sous Journaux du cluster, cochez la case Publier les journaux spécifiques au cluster sur Amazon S3. Remplacez la valeur de l'emplacement Amazon S3 par le compartiment Amazon S3 que vous avez créé, suivi de **`/logs`**. Par exemple, **`s3://DOC-EXAMPLE-BUCKET/logs`**. L'ajout de **`/logs`** crée un nouveau dossier appelé « journaux » dans votre compartiment, dans lequel Amazon EMR peut copier les fichiers journaux de votre cluster.
7. Sous Configuration de sécurité et autorisations, choisissez votre paire de clés EC2. Dans la même section, sélectionnez le menu déroulant Service role for Amazon EMR et choisissez EMR_. DefaultRole Sélectionnez ensuite le menu déroulant du rôle IAM pour le profil d'instance et choisissez EMR_EC2_. DefaultRole
8. Choisissez Créer un cluster pour lancer le cluster et ouvrir la page des détails du cluster.
9. Trouvez le statut du cluster à côté du nom du cluster. L'état passe de En cours de démarrage à En cours d'exécution, puis à En attente, à mesure qu'Amazon EMR approvisionne le cluster. Vous devrez peut-être choisir l'icône d'actualisation sur la droite ou actualiser votre navigateur pour voir les mises à jour de l'état.

L'état de votre cluster passe à En attente lorsque le cluster est opérationnel et prêt à accepter du travail. Pour plus d'informations sur la lecture d'un résumé de cluster, consultez Afficher l'état et les détails d'un cluster. Pour plus d'informations sur l'état du cluster, consultez Présentation du cycle de vie du cluster.

CLI

Pour lancer un cluster sur lequel Spark est installé avec AWS CLI

1. Créez des rôles IAM par défaut que vous pouvez ensuite utiliser pour créer votre cluster à l'aide de la commande suivante.

```
aws emr create-default-roles
```

Pour plus d'informations sur `create-default-roles`, consultez la rubrique Référence des commandes de la AWS CLI.

2. Créez un cluster Spark à l'aide de la commande suivante. Saisissez le nom de votre cluster à l'aide de l'option `--name` et indique le nom de votre paire de clés EC2 à l'aide de l'option `--ec2-attributes`.

```
aws emr create-cluster \
--name "<My First EMR Cluster>" \
--release-label <emr-5.36.2> \
--applications Name=Spark \
--ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-type m5.xlarge \
--instance-count 3 \
--use-default-roles
```

Notez les autres valeurs requises pour `--instance-type`, `--instance-count` et `--use-default-roles`. Ces valeurs ont été choisies pour les clusters à usage général. Pour plus d'informations sur `create-cluster`, consultez la rubrique Référence des commandes de la AWS CLI.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

Vous devriez voir une sortie semblable à la suivante. Le résultat généré indique le `ClusterId` et le `ClusterArn` de votre nouveau cluster. Notez votre `ClusterId`. Vous utilisez le `ClusterId` pour vérifier l'état du cluster et pour soumettre des travaux.

```
{
    "ClusterId": "myClusterId",
    "ClusterArn": "myClusterArn"
}
```

3. Vérifiez l'état de votre cluster à l'aide de la commande suivante.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Vous devriez voir un résultat comme le suivant avec l'objet `Status` de votre nouveau cluster.

```
{
    "Cluster": {
        "Id": "myClusterId",
        "Name": "My First EMR Cluster",
        "Status": {
            "State": "STARTING",
            "StateChangeReason": {
                "Message": "Configuring cluster software"
            }
        }
     }
}
```

La valeur `State` passe de `STARTING` à `RUNNING`, puis à `WAITING`, lorsqu'Amazon EMR approvisionne le cluster.

L'état du cluster passe à **`WAITING`** lorsque le cluster est opérationnel et prêt à accepter du travail. Pour plus d'informations sur l'état du cluster, consultez Présentation du cycle de vie du cluster.

Étape 2 : Gérer votre cluster Amazon EMR

Soumission de travail à Amazon EMR

Après avoir lancé un cluster, vous pouvez soumettre du travail au cluster en cours d'exécution pour traiter et analyser des données. Vous soumettez votre travail à un cluster Amazon EMR en tant qu'étape. L'étape est une unité de travail composée d'une ou plusieurs actions. Par exemple, vous pouvez soumettre une étape pour calculer des valeurs ou pour transférer et traiter des données. Vous pouvez soumettre des étapes lors de la création d'un cluster ou à un cluster en cours d'exécution. Dans cette partie du didacticiel, vous soumettez `health_violations.py` en tant qu'étape à votre cluster en cours d'exécution. Pour en savoir plus sur les étapes, consultez Soumission de travail à un cluster.

Console

Pour soumettre une application Spark en tant qu'étape par le biais de la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Dans le volet de navigation de gauche, sous EMR on EC2, choisissez Clusters, puis sélectionnez le cluster dans lequel vous souhaitez soumettre du travail. L'état du cluster doit être En attente.
3. Sélectionnez l'onglet Étapes, puis sélectionnez Ajouter une étape.
4. Configurez l'étape en fonction des consignes suivantes :

   - Pour Type, choisissez Application Spark. Vous devriez voir des champs supplémentaires pour le mode de déploiement, l'emplacement de l'application et les options Spark-submit.
   - Dans le champ Nom, saisissez un nouveau nom. Si vous avez de nombreuses étapes dans un cluster, le fait de nommer chaque étape vous aide à en garder la trace.
   - Pour le mode de déploiement, laissez la valeur par défaut Mode cluster. Pour plus d'informations sur les modes de déploiement de Spark, consultez la rubrique Présentation du mode cluster dans la documentation Apache Spark.
   - Dans le champ Emplacement de l'application, saisissez l'emplacement de votre script `health_violations.py` dans Amazon S3, par exemple *`s3://DOC-EXAMPLE-BUCKET/health_violations.py`*.
   - Laissez le champ des options Spark-submit vide. Pour plus d'informations sur les options `spark-submit`, consultez la rubrique Lancement d'applications à l'aide de spark-submit.
   - Dans le champ Arguments, saisissez les arguments et les valeurs suivants :

     ```
     --data_source s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv
     --output_uri s3://DOC-EXAMPLE-BUCKET/myOutputFolder
     ```

     Remplacez *`s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv`* par l'URI du compartiment S3 des données d'entrée que vous avez préparées dans Préparation d'une application avec des données d'entrée pour Amazon EMR.

     Remplacez *`DOC-EXAMPLE-BUCKET`* par le nom du bucket que vous avez créé pour ce didacticiel et par le nom du *`myOutputFolder`*dossier de sortie de votre cluster.

- Pour l'action en cas d'échec de l'étape, acceptez l'option par défaut Continuer. Ainsi, en cas d'échec de l'étape, le cluster continue de fonctionner.

5. Choisissez Ajouter pour soumettre l'étape. L'étape devrait apparaître dans la console avec l'état En attente.
6. Surveillez l'état de l'étape. Il devrait passer de En attente à En cours d'exécution, puis à Terminé. Pour actualiser l'état dans la console, cliquez sur l'icône d'actualisation à droite de Filtre. L'exécution du script prend environ une minute. Lorsque l'état devient Terminé, l'étape s'est achevée avec succès.

CLI

Pour soumettre une application Spark en tant qu'étape par le AWS CLI

1. Assurez-vous d'avoir le `ClusterId` du cluster que vous avez lancé dans Lancement d'un cluster Amazon EMR. Vous pouvez également récupérer l'identifiant de votre cluster à l'aide de la commande suivante.

```
aws emr list-clusters --cluster-states WAITING
```

2. Soumettez `health_violations.py` en tant qu'étape à l'aide de la commande `add-steps` et de votre `ClusterId`.

   - Vous pouvez donner un nom à votre étape en remplaçant *« Mon application Spark »*. Dans le tableau `Args`, remplacez *s3://DOC-EXAMPLE-BUCKET/health_violations.py* par l'emplacement de votre application `health_violations.py`.
   - Remplacez *s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv* par l'emplacement S3 de votre jeu de données `food_establishment_data.csv`.
   - Remplacez *s3://DOC-EXAMPLE-BUCKET/ MyOutputFolder* par le chemin S3 du compartiment que vous avez désigné et par le nom du dossier de sortie de votre cluster.
   - `ActionOnFailure=CONTINUE` signifie que le cluster continue à s'exécuter si l'étape échoue.

```
aws emr add-steps \
--cluster-id <myClusterId> \
```

```
--steps Type=Spark,Name="<My Spark
 Application>",ActionOnFailure=CONTINUE,Args=[<s3://DOC-EXAMPLE-
BUCKET/health_violations.py>,--data_source,<s3://DOC-EXAMPLE-BUCKET/
food_establishment_data.csv>,--output_uri,<s3://DOC-EXAMPLE-BUCKET/
MyOutputFolder>]
```

Pour plus d'informations sur la soumission d'étapes à l'aide de la CLI, consultez la Référence de commandes de la AWS CLI.

Une fois que vous avez soumis l'étape, vous devriez voir un résultat comme le suivant, avec une liste de `StepIds`. Puisque vous n'avez soumis qu'une seule étape, vous ne verrez qu'un seul identifiant dans la liste. Copiez votre identifiant d'étape. Utilisez votre identifiant d'étape pour vérifier l'état de l'étape.

```
{
    "StepIds": [
        "s-1XXXXXXXXXXA"
    ]
}
```

3. Vérifiez l'état de votre étape à l'aide de la commande `describe-step`.

```
aws emr describe-step --cluster-id <myClusterId> --step-id <s-1XXXXXXXXXXA>
```

Vous devriez voir un résultat comme le suivant, avec des informations sur votre étape.

```
{
    "Step": {
        "Id": "s-1XXXXXXXXXXA",
        "Name": "My Spark Application",
        "Config": {
            "Jar": "command-runner.jar",
            "Properties": {},
            "Args": [
                "spark-submit",
                "s3://DOC-EXAMPLE-BUCKET/health_violations.py",
                "--data_source",
                "s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv",
                "--output_uri",
                "s3://DOC-EXAMPLE-BUCKET/myOutputFolder"
```

```
            ]
        },
        "ActionOnFailure": "CONTINUE",
        "Status": {
            "State": "COMPLETED"
        }
    }
}
```

La valeur `State` de l'étape passe de `PENDING` à `RUNNING`, puis à `COMPLETED`, au fur et à mesure que l'étape s'exécute. L'exécution de l'étape prend environ une minute, il se peut donc que vous deviez vérifier l'état à plusieurs reprises.

Vous saurez que l'étape s'est terminée correctement lorsque le `State` passe à **`COMPLETED`**.

Pour plus d'informations sur le cycle de vie des étapes, consultez Exécuter des étapes pour traiter des données.

Affichage des résultats

Une fois qu'une étape s'est exécutée avec succès, vous pouvez consulter ses résultats dans votre dossier de sortie Amazon S3.

Affichage des résultats de **`health_violations.py`**

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment, puis le dossier de sortie que vous avez spécifié lorsque vous avez soumis l'étape. Par exemple, *`DOC-EXAMPLE-BUCKET`*, puis. *`myOutputFolder`*
3. Vérifiez que les éléments suivants apparaissent dans votre dossier de sortie :
   - Un objet de petite taille appelé `_SUCCESS`.
   - Un fichier CSV commençant par le préfixe `part-` qui contient vos résultats.
4. Choisissez l'objet contenant vos résultats, puis choisissez Télécharger pour enregistrer les résultats dans votre système de fichiers local.
5. Ouvrez le résultats dans l'éditeur de votre choix. Le fichier de sortie répertorie les dix établissements ayant enregistré le plus grand nombre d'infractions rouges. Le fichier de sortie indique également le nombre total d'infractions rouges pour chaque établissement.

Voici un exemple de résultats `health_violations.py`.

```
name, total_red_violations
SUBWAY, 322
T-MOBILE PARK, 315
WHOLE FOODS MARKET, 299
PCC COMMUNITY MARKETS, 251
TACO TIME, 240
MCDONALD'S, 177
THAI GINGER, 153
SAFEWAY INC #1508, 143
TAQUERIA EL RINCONSITO, 134
HIMITSU TERIYAKI, 128
```

Pour plus d'informations sur la sortie des clusters Amazon EMR, consultez Configuration d'un emplacement de sortie.

(Facultatif) Connexion à votre cluster Amazon EMR en cours d'exécution

Lorsque vous utilisez Amazon EMR, vous souhaiterez peut-être vous connecter à un cluster en cours d'exécution pour lire les fichiers journaux, déboguer le cluster ou utiliser des outils de la CLI tels que le shell de Spark. Amazon EMR vous permet de vous connecter à un cluster à l'aide du protocole Secure Shell (SSH). Cette section explique comment configurer SSH, vous connecter à votre cluster et consulter les fichiers journaux de Spark. Pour plus d'informations sur la connexion à un cluster, consultez Authentification auprès des nœuds de cluster Amazon EMR.

Autorisation des connexions SSH à votre cluster

Avant de vous connecter à votre cluster, vous devez modifier les groupes de sécurité de votre cluster pour autoriser les connexions SSH entrantes. Les groupes de sécurité Amazon EC2 agissent en tant que pares-feux virtuels pour contrôler le trafic entrant et sortant de votre cluster. Lorsque vous avez créé votre cluster pour ce didacticiel, Amazon EMR a créé les groupes de sécurité suivants en votre nom :

ElasticMapReduce-maître

Le groupe de sécurité gérés Amazon EMR par défaut, associé au nœud primaire. Dans un cluster Amazon EMR, le nœud primaire est une instance Amazon EC2 qui gère le cluster.

ElasticMapReduce-esclave

Groupe de sécurité par défaut associé aux nœuds principaux et aux nœuds de tâches.

Console

Pour autoriser l'accès SSH aux sources fiables pour le groupe de sécurité principal avec la console

Pour modifier vos groupes de sécurité, vous devez avoir l'autorisation de gérer les groupes de sécurité pour le VPC dans lequel se trouve le cluster. Pour plus d'informations, consultez Modification des autorisations d'un utilisateur et l'exemple de politique permettant de gérer les groupes de sécurité EC2 dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Dans le volet de navigation de gauche, sous EMR on EC2, choisissez Clusters, puis le cluster que vous souhaitez mettre à jour. La page de détails du cluster s'ouvre. L'onglet Propriétés de cette page devrait être présélectionné.
3. Sous Mise en réseau dans l'onglet Propriétés, sélectionnez la flèche à côté des groupes de sécurité EC2 (pare-feu) pour développer cette section. Sous Nœud primaire, sélectionnez le lien du groupe de sécurité. Lorsque vous avez effectué les étapes suivantes, vous avez la possibilité de revenir à cette étape, de sélectionner les Nœuds principaux et de tâche, et de répéter les étapes suivantes pour permettre l'accès du client SSH aux nœuds principaux et de tâche.
4. Ceci ouvre la console EC2. Sélectionnez l'onglet Règles entrantes, puis Modifier les règles entrantes.
5. Vérifiez s'il existe une règle entrante qui autorise l'accès public avec les paramètres suivants. Si elle existe, choisissez Supprimer pour la supprimer.

   - Type

     SSH
   - Port

     22
   - Source

Personnalisé 0.0.0.0/0

> **Warning**
>
> Avant décembre 2020, le groupe de sécurité ElasticMapReduce -master disposait d'une règle préconfigurée pour autoriser le trafic entrant sur le port 22 en provenance de toutes les sources. Cette règle a été créée pour simplifier les connexions SSH initiales au nœud principal. Nous vous recommandons vivement de supprimer cette règle d'entrée et de limiter le trafic aux sources fiables.

6. Faites défiler la liste des règles jusqu'en bas et sélectionnez Ajouter une règle.
7. Dans le champ Type, sélectionnez SSH. En sélectionnant SSH, vous saisissez automatiquement TCP pour le protocole et 22 pour la plage de ports.
8. Pour source, sélectionnez Mon adresse IP pour ajouter automatiquement votre adresse IP en tant qu'adresse source. Vous pouvez également ajouter une plage d'adresses IP de clients fiables personnalisées ou créer des règles supplémentaires pour d'autres clients. De nombreux environnements réseau allouent des adresses IP de manière dynamique. Il se peut donc que vous deviez mettre à jour vos adresses IP pour les clients fiables à l'avenir.
9. Choisissez Enregistrer.
10. Choisissez éventuellement les nœuds principaux et de tâches dans la liste et répétez les étapes ci-dessus pour autoriser l'accès du client SSH aux nœuds principaux et aux nœuds de tâches.

Old console

Pour accorder à des sources fiables un accès SSH au groupe de sécurité principal via la console

Pour modifier vos groupes de sécurité, vous devez avoir l'autorisation de gérer les groupes de sécurité pour le VPC dans lequel se trouve le cluster. Pour plus d'informations, consultez Modification des autorisations d'un utilisateur et l'exemple de politique permettant de gérer les groupes de sécurité EC2 dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Clusters. Choisissez l'ID du cluster que vous souhaitez modifier.

3. Dans le volet Réseau et sécurité, développez la liste déroulante des groupes de sécurité EC2 (pare-feu).
4. Sous Nœud principal, choisissez votre groupe de sécurité.
5. Choisissez Modifier les règles entrantes.
6. Vérifiez s'il existe une règle entrante qui autorise l'accès public avec les paramètres suivants. Si elle existe, choisissez Supprimer pour la supprimer.

   - Type

     SSH
   - Port

     22
   - Source

     Personnalisé 0.0.0.0/0

   > **Warning**
   >
   > Avant décembre 2020, une règle préconfigurée autorisait le trafic entrant sur le port 22 en provenance de toutes les sources. Cette règle a été créée pour simplifier les connexions SSH initiales au nœud primaire. Nous vous recommandons vivement de supprimer cette règle d'entrée et de limiter le trafic aux sources fiables.

7. Faites défiler la liste des règles jusqu'en bas et sélectionnez Ajouter une règle.
8. Dans le champ Type, sélectionnez SSH.

   En sélectionnant SSH, vous saisissez automatiquement TCP pour le protocole et 22 pour la plage de ports.
9. Pour source, sélectionnez Mon adresse IP pour ajouter automatiquement votre adresse IP en tant qu'adresse source. Vous pouvez également ajouter une plage d'adresses IP de clients fiables personnalisées ou créer des règles supplémentaires pour d'autres clients. De nombreux environnements réseau allouent des adresses IP de manière dynamique. Il se peut donc que vous deviez mettre à jour vos adresses IP pour les clients fiables à l'avenir.
10. Choisissez Enregistrer.

11. Choisissez éventuellement l'autre groupe de sécurité sous Nœuds principaux et de tâches dans le volet Réseau et sécurité et répétez les étapes ci-dessus pour autoriser l'accès du client SSH aux nœuds principaux et aux nœuds de tâches.

Connectez-vous à votre cluster à l'aide du AWS CLI

Quel que soit votre système d'exploitation, vous pouvez créer une connexion SSH à votre cluster à l'aide de la AWS CLI.

Pour vous connecter à votre cluster et consulter les fichiers journaux à l'aide du AWS CLI

1. Utilisez la commande suivante pour ouvrir une connexion SSH à votre cluster. Remplacez *<mykeypair.key>* par le chemin d'accès complet et le nom du fichier de votre paire de clés. Par exemple, `C:\Users\<username>\.ssh\mykeypair.pem`.

   ```
   aws emr ssh --cluster-id <j-2AL4XXXXXX5T9> --key-pair-file <~/mykeypair.key>
   ```

2. Naviguez vers `/mnt/var/log/spark` pour accéder aux journaux Spark sur le nœud principal de votre cluster. Affichez ensuite les fichiers qui se trouvent à cet emplacement. Pour obtenir la liste des fichiers journaux supplémentaires sur le nœud principal, consultez Affichage des fichiers journaux sur le nœud primaire.

   ```
   cd /mnt/var/log/spark
   ls
   ```

Étape 3 : Nettoyer vos ressources Amazon EMR

Arrêt de votre cluster

Maintenant que vous avez soumis du travail à votre cluster et que vous avez consulté les résultats de votre PySpark application, vous pouvez mettre fin au cluster. L'arrêt du cluster arrête toutes les charges Amazon EMR et les instances Amazon EC2 qui lui sont associées.

Lorsque vous arrêtez un cluster, Amazon EMR conserve gratuitement les métadonnées relatives au cluster pendant deux mois. Les métadonnées archivées vous permettent de cloner le cluster pour une nouvelle tâche ou de retenir la configuration du cluster à des fins de référence. Les métadonnées n'incluent pas les données que le cluster écrit dans S3, ni les données stockées dans HDFS sur le cluster.

> **Note**
>
> La console Amazon EMR ne vous permet pas de supprimer un cluster de la vue de la liste après avoir arrêté le cluster. Le cluster arrêté disparaît de la console lorsqu'Amazon EMR efface ses métadonnées.

Console

Pour terminer le cluster à l'aide de la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Clusters, puis sélectionnez le cluster que vous voulez arrêter.
3. Dans le menu déroulant Actions, choisissez Arrêter le cluster.
4. Dans la boîte de dialogue, choisissez Arrêter. Selon la configuration du cluster, l'arrêt peut prendre de 5 à 10 minutes. Pour plus d'informations sur la création de clusters Amazon EMR, consultez Arrêter un cluster.

CLI

Pour terminer le cluster à l'aide du AWS CLI

1. Lancez le processus d'arrêt du cluster à l'aide de la commande suivante. Remplacez *< myClusterId >* par l'ID de votre cluster d'échantillons. La commande ne renvoie pas de résultat.

```
aws emr terminate-clusters --cluster-ids <myClusterId>
```

2. Pour vérifier que le processus d'arrêt du cluster est en cours, vérifiez l'état du cluster à l'aide de la commande suivante.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Voici un exemple de résultat au format JSON. Le `Status` du cluster doit passer de **`TERMINATING`** à **`TERMINATED`**. Selon la configuration de votre cluster, l'arrêt peut prendre

de 5 à 10 minutes. Pour plus d'informations sur l'arrêt d'un cluster Amazon EMR, consultez Arrêter un cluster.

```
{
    "Cluster": {
        "Id": "j-xxxxxxxxxxxx",
        "Name": "My Cluster Name",
        "Status": {
            "State": "TERMINATED",
            "StateChangeReason": {
                "Code": "USER_REQUEST",
                "Message": "Terminated by user request"
            }
        }
    }
}
```

Suppression des ressources S3

Pour éviter des frais supplémentaires, vous devez supprimer votre compartiment Amazon S3. La suppression du compartiment entraîne la suppression de toutes les ressources Amazon S3 pour ce didacticiel. Votre compartiment doit contenir :

- Le PySpark script
- Le jeu de données d'entrée
- Votre dossier de résultats de sortie
- Votre dossier de fichiers journaux

Vous devrez peut-être prendre des mesures supplémentaires pour supprimer les fichiers stockés si vous avez enregistré votre PySpark script ou votre sortie dans un autre emplacement.

> **Note**
>
> Votre cluster doit être arrêté avant que vous ne supprimiez votre compartiment. Sinon, vous risquez de ne pas être autorisé à vider le compartiment.

Pour supprimer votre compartiment, suivez les instructions de la rubrique Comment supprimer un compartiment S3 ? dans le Guide de l'utilisateur Amazon Simple Storage Service.

Étapes suivantes

Vous venez de lancer votre premier cluster Amazon EMR du début à la fin. Vous avez également effectué des tâches EMR essentielles telles que la préparation et la soumission des applications de big data, la visualisation des résultats et l'arrêt d'un cluster.

Consultez les rubriques suivantes pour en savoir plus sur la manière dont vous pouvez personnaliser votre flux de travail Amazon EMR.

Découvrez les applications de big data pour Amazon EMR

Découvrez et comparez les applications de big data que vous pouvez installer sur un cluster dans le Guide de mise à jour d'Amazon EMR. Le guide de mise à jour détaille chaque version EMR et comprend des conseils pour l'utilisation d'environnements tels que Spark et Hadoop sur Amazon EMR.

Planification du matériel, de la mise en réseau et de la sécurité du cluster

Dans ce didacticiel, vous avez créé un cluster EMR simple sans configurer d'options avancées. Les options avancées vous permettent de spécifier les types d'instances Amazon EC2, le réseau du cluster et la sécurité du cluster. Pour plus d'informations sur la planification et le lancement d'un cluster répondant à vos besoins, consultez Planification et configuration des clusters et Sécurité dans Amazon EMR.

Gestion des clusters

Approfondissez l'utilisation de clusters en cours d'exécution dans Gestion des clusters. Pour gérer un cluster, vous pouvez vous connecter au cluster, effectuer les étapes de débogage et suivre les activités et l'état du cluster. Vous pouvez également ajuster les ressources du cluster en fonction des demandes de charge de travail grâce à la mise à l'échelle gérée par EMR.

Utilisation d'une interface différente

Outre la console Amazon EMR, vous pouvez gérer Amazon EMR à l'aide de l'API du AWS Command Line Interface service Web ou de l'un des nombreux SDK pris en charge. AWS Pour plus d’informations, consultez Interfaces de gestion.

Vous pouvez également interagir avec les applications installées sur les clusters Amazon EMR de plusieurs façons. Certaines applications, comme Apache Hadoop, publient des interfaces web que vous pouvez consulter. Pour plus d'informations, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Consultation du blog technique EMR

Pour des exemples de démonstration et des discussions techniques approfondies sur les nouvelles fonctionnalités d'Amazon EMR, consultez le blog AWS big data.

Console Amazon EMR

La console propose une interface mise à jour qui vous permet de gérer de manière intuitive votre environnement Amazon EMR et d'accéder facilement à la documentation, aux informations sur les produits et à d'autres ressources.

Fonctionnalités de la console

La console Amazon EMR est disponible à l'adresse URL suivante :

- URL de la console — https://console.aws.amazon.com/emr

Le tableau suivant répertorie l'état des principaux composants de la console Amazon EMR.

Composant de la console Amazon EMR	Console	
EMR Studio	✓	
Création et gestion de clusters	✓	
Blocage de l'accès public	✓	
Surveillez les CloudWatch événements Amazon	✓	
Configurations de la sécurité	✓	
Clusters virtuels (Amazon EMR sur EKS)	✓	
Afficher et gérer vos sous-réseaux Amazon Virtual Private Cloud [1]	✓	
Carnets de notes (2)	✓	

[1] Dans la console, vous pouvez afficher et gérer vos sous-réseaux Amazon VPC dans la section Mise en réseau lorsque vous créez un cluster.

[2] notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Résumé des différences

Cette section décrit les fonctionnalités de l'expérience de la console Amazon EMR. Ces fonctionnalités se répartissent dans les catégories suivantes :

- Compatibilité des clusters dans la console
- Création de clusters
- Afficher ou modifier les détails du cluster
- Affichage et recherche de clusters
- Différences lorsque vous travaillez avec des configurations de sécurité

Compatibilité des clusters dans la console

Dans certains cas, un cluster que vous avez créé peut ne pas être compatible avec la console. La liste suivante décrit les exigences de compatibilité pour la console Amazon EMR.

- La console prend en charge les clusters créés dans les versions 5.20.1 et ultérieures d'Amazon EMR.
- Vous pouvez cloner des clusters qui utilisent le dimensionnement automatique dans la console, mais vous ne pouvez créer de nouveaux clusters que si vous souhaitez les redimensionner manuellement ou utiliser le dimensionnement géré.

Pour créer et utiliser des clusters des versions 5.20.1 et antérieures, vous pouvez utiliser le AWS Command Line Interface (AWS CLI) ou le AWS SDK.

Création de clusters

Capacité	Console	

Capacité	Console	
Terminologie : types de nœuds de cluster Amazon EMR	Principal, tâche	
Versions Amazon EMR prises en charge[1]	Amazon EMR 5.20.1 et versions ultérieures	
Lancer rapidement un cluster	Cliquez sur le bouton Créer un cluster situé sous le panneau Résumé. Le nom de votre cluster ne peut pas contenir les caractère s <, >, $, | ou `(backtick).	
Configurer un délai d'expirat ion de l'allocation Spot	Définissez un délai d'expirat ion pour allouer des instances pour chaque flotte de votre cluster.	
Fonctions de service et rôle de profil d'instance Amazon EC2	La console ne crée pas de rôles par défaut ; vous devez créer des rôles avec la console IAM ou sélection ner un rôle IAM déjà créé	
Visibilité des clusters	Depuis la console Amazon EMR, vous ne pouvez pas rendre un cluster visible par tous les utilisate urs ; votre politique IAM détermine l'accès au cluster	

Capacité	Console	
Réseaux : configuration de sous-réseaux privés	Vous devez configurer les points de terminaison Amazon S3 et les passerelles NAT à partir de leurs consoles Amazon S3 et Amazon VPC respectives	
Vue cohérente du système de fichiers EMR (EMRFS CV)	Avec la sortie d'Amazon S3 Strong read-after-write Cohérence le 1er décembre 2020, vous n'avez pas besoin d'utiliser EMRFS CV avec vos clusters EMR	
Débogage	Vous pouvez déboguer des tâches à l'aide de l'interface utilisateur de l'application sur la page de détails du cluster	

[1] Vous ne pouvez pas créer ou modifier de clusters à l'aide de versions antérieures à Amazon EMR 5.20.1 dans la console, mais tous les clusters existants créés à l'aide de versions antérieures à la version 5.20.1 continueront de fonctionner. Pour créer et modifier des clusters avec des versions d'Amazon EMR antérieures à la version 5.20.1, utilisez l'API ou la CLI. Vous pouvez afficher tous les clusters à l'aide de la console, mais les consoles créées avant la version 5.20.1 peuvent ne pas être compatibles avec les nouvelles fonctionnalités.

Affichage et recherche de clusters

Le tableau suivant explique comment utiliser la console Amazon EMR pour afficher, visualiser et rechercher des clusters.

> **Note**
>
> L'application d'un filtre de données à la liste des clusters interroge l'ensemble de la base de données. En revanche, lorsque vous saisissez une chaîne de texte dans le champ de recherche, la recherche ne s'applique qu'aux résultats que la liste a chargés côté client.

Capacité	Console	
Affichage des détails de cluster	Vous pouvez sélectionner l'ID de cluster pour afficher les détails complets du cluster, tels que les options de configuration, les interfaces utilisateur persistantes des applications et les journaux.	
Recherche de clusters	Utilisez un champ de recherche unique pour saisir des requêtes de recherche textuelles et pour créer et appliquer des filtres de données tels que « Statut = Tout statut actif ».	
Trouver des clusters défaillants	Pour rechercher des clusters en échec, appliquez le filtre Status = Terminé avec des erreurs.	

Afficher ou modifier les détails du cluster

Capacité	Console	

Capacité	Console	
Affichage des instances de vos groupes d'instances et de vos flottes d'instances, ainsi que des options de mise à l'échelle, d'allocation, de redimensionnement et de résiliation	Consultez les options et les détails des instances dans l'onglet Instances. Consultez les options de résiliation dans l'onglet Propriétés.	
Affichage des interfaces utilisateur, des journaux et des configurations d'applications (Interface utilisateur Apache Spark, service d'historique Spark, interface utilisateur Apache Tez, serveur de chronologie YARN)	Consultez les configurations de cluster dans l'onglet Configurations. Lancez une interface utilisateur d'application active et persistante pour consulter les journaux d'une application depuis l'onglet Applications.	
Exportation d'un cluster vers la CLI	Option disponible dans les menus Actions de détails de cluster et d'affichage de liste avec l'intitulé « Commande d'affichage pour le clonage d'un cluster »	

Différences lorsque vous travaillez avec des configurations de sécurité

Capacité	Console	
Clonage de configurations de sécurité	✓	

Capacité	Console	
Gouvernance fédérée à l'aide de Trino et Apache Ranger	✓	
Utilisation d'un rôle d'exécution pour soumettre une tâche à un cluster[1]	✓	
Autorisation de l'accès aux données de système de fichiers EMR (EMRFS)	Points d'accès Amazon S3	
AWS Lake Formation contrôles d'accès	Rôles d'exécution	

[1] Pour transmettre un rôle lors de la soumission d'une étape, votre cluster doit utiliser une configuration de sécurité associée à une politique d'autorisations IAM afin que l'utilisateur ne puisse transmettre que les rôles approuvés et que vos tâches puissent accéder aux ressources Amazon EMR. Pour plus d'informations, voir Rôles d'exécution pour les étapes Amazon EMR.

Amazon EMR Studio

Amazon EMR Studio est un environnement de développement intégré (IDE) basé sur le Web pour les blocs-notes Jupyter entièrement gérés qui s'exécutent sur des clusters Amazon EMR. Vous pouvez configurer un studio EMR pour que votre équipe développe, visualise et débogue des applications écrites en R, Python, Scala et. PySpark EMR Studio est intégré à AWS Identity and Access Management (IAM) et à IAM Identity Center afin que les utilisateurs puissent se connecter à l'aide de leurs informations d'identification d'entreprise.

Vous pouvez créer gratuitement un studio EMR. Les frais afférents au stockage Amazon S3 et aux clusters Amazon EMR s'appliquent lorsque vous utilisez EMR Studio. Pour connaître les détails et les points forts du produit, consultez la page consacrée au service Amazon EMR Studio.

Principales fonctionnalités d'EMR Studio

Amazon EMR Studio offre les fonctionnalités suivantes :

- Authentification des utilisateurs avec AWS Identity and Access Management (IAM), AWS IAM Identity Center avec ou sans propagation d’identité approuvée et votre fournisseur d’identité d’entreprise.
- Accès et lancement de clusters Amazon EMR à la demande pour l’exécution de tâches sur les blocs-notes Jupyter.
- Connexion à Amazon EMR sur les clusters EKS pour soumettre les tâches au fur et à mesure de leur exécution.
- Exploration et enregistrement d'exemples de blocs-notes. Pour plus d'informations sur les exemples de blocs-notes, consultez le référentiel d'exemples de GitHub blocs-notes EMR Studio.
- Analysez les données à l'aide de Python PySpark, Spark Scala, Spark R ou SparkSQL, et installez des noyaux et des bibliothèques personnalisés.
- Collaboration en temps réel avec d'autres utilisateurs dans le même espace de travail. Pour plus d’informations, consultez Configuration de la collaboration dans Workspace.
- Utilisation de l'explorateur SQL d'EMR Studio pour parcourir votre catalogue de données, exécuter des requêtes SQL et télécharger les résultats avant de travailler avec les données dans un bloc-notes.
- Exécution de blocs-notes paramétrés dans le cadre de flux de travail planifiés à l'aide d'un outil d'orchestration tel qu'Apache Airflow ou Amazon Managed Workflows for Apache Airflow. Pour plus

d'informations, consultez la rubrique Orchestration des tâches d'analyse sur les blocs-notes EMR à l'aide de MWAA sur le blog AWS Big Data.

- Référentiels de codes de liens tels que GitHub et. BitBucket
- Suivi et débogage des tâches à l'aide du serveur d'historique Spark, de l'interface utilisateur Tez ou du serveur de chronologie YARN.

EMR Studio est également éligible HIPAA et est certifié HITRUST CSF et SOC 2. Pour plus d'informations sur la conformité à la loi HIPAA des services AWS, consultez https://aws.amazon.com/compliance/hipaa-compliance/. Pour en savoir plus sur la conformité à HITRUST CSF des services AWS, consultez https://aws.amazon.com/compliance/hitrust/. Pour plus d'informations sur les autres programmes de conformité des services AWS, consultez la rubrique Services AWS concernés par le programme de conformité.

Historique des fonctionnalités d'Amazon EMR Studio

Ce tableau répertorie les mises à jour apportées à la fonctionnalité de mise à l'échelle gérée par Amazon EMR.

Date de publication	Capacité
5 janvier 2024	Ajout du support pour EMR Studio en AWS GovCloud (USA Est) et AWS GovCloud (USA Ouest).
26 novembre 2023	Ajout de la prise en charge de la propagation d'identité approuvée pour EMR Studio avec l'authentification IAM Identity Center.
26 octobre 2023	Ajout de la possibilité de créer une application EMR sans serveur dotée d'une fonctionnalité interactive.
28 février 2023	Ajout de la prise en charge des clés AWS KMS gérées par le client pour le stockage des journaux des applications EMR sans serveur.
23 février 2023	Ajout de la création de rôles IAM en un clic pour la soumission de tâches EMR sans serveur. Ajout de la recherche ECR lorsque vous sélectionnez une image personnalisée pour les applications EMR sans serveur.

Date de publication	Capacité
27 janvier 2023	Les blocs-notes d'exécution sans tête peuvent suivre la progressi on de l'exécution de chaque cellule à l'aide de la commande magique `%execute_notebook` .
23 janvier 2023	Les applications persistantes ont été optimisées pour un lancement plus rapide.

Comment fonctionne Amazon EMR Studio

Un Studio Amazon EMR est une ressource Amazon EMR que vous créez pour une équipe d'utilisateurs. Chaque studio est un environnement de développement autonome, intégré et basé sur le Web pour les blocs-notes Jupyter qui s'exécutent sur des clusters Amazon EMR. Les utilisateurs se connectent à un studio avec leurs informations d'identification d'entreprise.

Chaque EMR Studio que vous créez utilise les ressources suivantes AWS :

- Cloud privé virtuel (VPC) Amazon avec des sous-réseaux : les utilisateurs exécutent des noyaux et des applications Studio sur Amazon EMR, et Amazon EMR sur des clusters EKS dans le VPC spécifié. Un EMR Studio peut se connecter à n'importe quel cluster dans les sous-réseaux que vous spécifiez lors de la création du studio.
- Rôles IAM et politiques d'autorisation : pour gérer les autorisations des utilisateurs, vous créez des politiques d'autorisations IAM que vous associez à l'identité IAM d'un utilisateur ou à un rôle d'utilisateur. EMR Studio utilise également un rôle de service IAM et des groupes de sécurité pour interagir avec d'autres services AWS. Pour plus d'informations, consultez Contrôle d'accès et Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio.
- Groupes de sécurité : EMR Studio utilise des groupes de sécurité pour établir un canal réseau sécurisé entre le studio et un cluster EMR.
- Emplacement de sauvegarde Amazon S3 : EMR Studio enregistre le travail du bloc-notes dans un emplacement Amazon S3.

Les étapes suivantes expliquent comment créer et administrer un EMR Studio :

1. Créez un studio dans votre environnement Compte AWS avec l'authentification IAM ou IAM Identity Center. Pour obtenir des instructions, consultez Configurer un Amazon EMR Studio.

2. Attribuez un utilisateur ou un groupe à votre EMR Studio Utilisez des politiques d'autorisation pour définir des autorisations précises pour chaque utilisateur. Pour de plus amples informations, consultez la rubrique Attribuer et gérer les utilisateurs d'EMR Studio.
3. Commencez à surveiller les actions d'EMR Studio à l'aide d'événements AWS CloudTrail. Pour de plus amples informations, veuillez consulter Surveiller les actions Amazon EMR Studio.
4. Offrez davantage d'options de cluster aux utilisateurs de Studio avec des modèles de clusters et Amazon EMR sur les points de terminaison gérés par EKS.

Authentification et connexion utilisateur

Amazon EMR Studio prend en charge deux modes d'authentification : le mode d'authentification IAM et le mode d'authentification IAM Identity Center. Le mode IAM utilise AWS Identity and Access Management (IAM), tandis que le mode IAM Identity Center utilise AWS IAM Identity Center. Lorsque vous créez un EMR Studio, vous choisissez le mode d'authentification pour tous les utilisateurs de ce studio.

Mode d'authentification IAM

Avec le mode d'authentification IAM, vous pouvez utiliser l'authentification IAM ou la fédération IAM.

L'authentification IAM vous permet de gérer les identités IAM telles que les utilisateurs, les groupes et les rôles dans IAM. Vous autorisez les utilisateurs à accéder à un studio avec des politiques d'autorisations IAM et un contrôle d'accès par attributs (ABAC).

La fédération IAM vous permet d'établir un lien de confiance entre un fournisseur d'identité (IdP) tiers et AWS afin que vous puissiez gérer les identités des utilisateurs par le biais de votre IdP.

Mode d'authentification IAM Identity Center

Le mode d'authentification IAM Identity Center vous permet d'accorder aux utilisateurs un accès fédéré à un EMR Studio. Vous pouvez utiliser IAM Identity Center pour authentifier les utilisateurs et les groupes à partir de votre répertoire IAM Identity Center, de votre annuaire d'entreprise existant ou d'un IdP externe tel qu'Azure Active Directory (AD). Vous gérez ensuite les utilisateurs avec votre fournisseur d'identité (IdP).

EMR Studio prend en charge l'utilisation des fournisseurs d'identité suivants pour IAM Identity Center :

- AWS Managed Microsoft AD et Active Directory autogéré : pour plus d'informations, consultez Se connecter à votre annuaire Microsoft AD.
- Fournisseurs basés sur le protocole SAML : pour une liste complète, consultez Fournisseurs d'identité pris en charge.
- L'annuaire IAM Identity Center : pour plus d'informations, consultez les rubrique Gestion des identités dans IAM Identity Center et Propagation d'identité approuvée entre les applications du Guide de l'utilisateur AWS IAM Identity Center.

Comment l'authentification affecte la connexion et l'attribution des utilisateurs

Le mode d'authentification que vous choisissez pour EMR Studio affecte la manière dont les utilisateurs se connectent à un studio, la manière dont vous attribuez un utilisateur à un studio et la manière dont vous autorisez (accordez des autorisations) les utilisateurs à effectuer des actions telles que la créer de nouveaux clusters Amazon EMR.

Le tableau suivant récapitule les méthodes de connexion à EMR Studio en fonction du mode d'authentification.

Options de connexion à EMR Studio par mode d'authentification

Mode d'authentification	Méthode de connexion	Description
• IAM (authentification et fédération) • IAM Identity Center	URL EMR Studio	Les utilisateurs se connectent à un studio à l'aide de l'URL d'accès au studio. Par exemple, `https://xxxxxxxxxxxxxxxxxxxxxxxx.emrstudio-prod.us-east-1.amazonaws.com` . Lorsque vous utilisez l'authentification IAM, les utilisateurs saisissent les informations d'identification IAM. Lorsque vous utilisez la fédération IAM ou IAM Identity Center, EMR Studio redirige les utilisateurs vers l'URL de connexion de votre fournisseur d'identité pour saisir les informations d'identification.

Mode d'authentification	Méthode de connexion	Description
		Dans le contexte de la fédération d'identité, cette option de connexion est appelée connexion initiée par le fournisseur de services (SP).
• IAM (fédération) • IAM Identity Center	Portail fournisseur d'identité (IdP)	Les utilisateurs se connectent au portail de votre fournisseur d'identité, tel que le portail Azure, et lancent la console Amazon EMR. Après avoir lancé la console Amazon EMR, les utilisateurs sélectionnent et ouvrent un studio dans la liste des studios. Vous pouvez également configurer EMR Studio en tant qu'application SAML afin que les utilisateurs puissent se connecter à un studio spécifique depuis le portail de votre fournisseur d'identité. Pour obtenir des instructions, consultez Configurer un EMR Studio en tant qu'application SAML sur votre portail IdP. Dans le contexte de la fédération d'identité, cette option de connexion est appelée connexion initiée par le fournisseur d'identité (IdP).
• IAM (authentification)	AWS Management Console	Les utilisateurs se connectent à AWS Management Console à l'aide des informations d'identification IAM et ouvrent un studio depuis la liste des studios de la console Amazon EMR.

Le tableau suivant décrit l'attribution et l'autorisation des utilisateurs pour EMR Studio par mode d'authentification.

Attribution et autorisation des utilisateurs EMR Studio par mode d'authentification

Mode d’authentification	Attribution d'utilisateurs	Autorisation utilisateur
IAM (authentification et fédération)	Autorisez l'action `CreateStudioPresignedUrl` dans une politique d'autorisations IAM rattachée à une identité IAM (utilisateur, groupe ou rôle). Pour les utilisateurs fédérés, autorisez l'action `CreateStudioPresignedUrl` dans un IAM au sein de la politique d'autorisations que vous configurez pour le rôle IAM de la fédération. Utilisez le contrôle d'accès par attributs (ABAC) pour spécifier le ou les studios auxquels l'utilisateur peut accéder. Pour obtenir des instructions, consultez Attribuer un utilisateur ou un groupe à un EMR Studio.	Définissez des politiques d'autorisation IAM qui autorisent certaines actions EMR Studio. Pour les utilisateurs natifs, rattachez la politique d'autorisations IAM à une identité IAM (utilisateur, groupe ou rôle). Pour les utilisateurs fédérés, autorisez des actions Studio au sein de la politique d'autorisations que vous configurez pour le rôle IAM de la fédération. Pour de plus amples informations, veuillez consulter Configurer les autorisations utilisateur d'EMR Studio pour Amazon EC2 ou Amazon EKS.
IAM Identity Center	Pour les Studios créés pour lesquels `IdCUserAssignment` est défini sur `REQUIRED`, associez les utilisateurs au Studio à une politique de session spécifiée. Pour de plus amples informations, veuillez consulter Attribuer un utilisateur ou un groupe à un EMR Studio.	Facultatif : définissez des politiques de session IAM qui autorisent certaines actions EMR Studio. Associez une politique de session à un utilisateur lorsque vous attribuez l'utilisateur à un studio. Pour de plus amples informations, veuillez consulter Autorisations utilisateur pour le mode d'authentification IAM Identity Center.

Mode d’authentification	Attribution d'utilisateurs	Autorisation utilisateur
	Pour les studios créés pour lesquels `IdCUserAssignment` est défini sur `OPTIONAL`, tous les utilisateurs ou groupes Identity Center peuvent accéder au Studio.	

Contrôle d'accès

Dans Amazon EMR Studio, vous configurez l'autorisation utilisateur avec des politiques basées sur l'identité (IAM) AWS Identity and Access Management. Dans ces politiques, vous spécifiez les actions et les ressources autorisées, ainsi que les conditions dans lesquelles les actions sont autorisées.

Autorisations utilisateur pour le mode d'authentification IAM

Pour définir les autorisations utilisateur lorsque vous utilisez l'authentification IAM pour EMR Studio, vous autorisez des actions telles que `elasticmapreduce:RunJobFlow` dans le cadre d'une politique d'autorisation IAM. Vous pouvez créer une ou plusieurs politiques d'autorisations à utiliser. Par exemple, vous pouvez créer une politique de base qui n'autorise pas un utilisateur à créer de nouveaux clusters Amazon EMR, et une autre politique qui autorise la création de clusters. Pour obtenir la liste des actions Studio, consultez Autorisations AWS Identity and Access Management pour les utilisateurs d'EMR Studio.

Autorisations utilisateur pour le mode d'authentification IAM Identity Center

Lorsque vous utilisez l'authentification IAM Identity Center, vous créez un rôle d'utilisateur EMR Studio unique. Le rôle d'utilisateur est un rôle IAM dédié qu'un studio assume lorsqu'un utilisateur se connecte.

Vous associez des politiques de session IAM au rôle d'utilisateur EMR Studio. Une politique de session est un type spécial de politique d'autorisation IAM qui limite ce qu'un utilisateur fédéré peut faire pendant une session de connexion à Studio. Les politiques de session vous permettent de définir des autorisations spécifiques pour un utilisateur ou un groupe sans créer plusieurs rôles d'utilisateur pour EMR Studio.

Lorsque vous attribuez des utilisateurs et des groupes à un studio, vous associez une politique de session à cet utilisateur ou à ce groupe afin d'appliquer des autorisations précises. Vous pouvez également mettre à jour la politique de session d'un utilisateur ou d'un groupe à tout moment. Amazon EMR stocke chaque association de politique de session que vous créez.

Pour plus d'informations sur les politiques de session, consultez Autorisations et politiques dans le Guide de l'utilisateur AWS Identity and Access Management.

Espaces de travail

Les Workspaces sont les principaux éléments constitutifs d'Amazon EMR Studio. Pour organiser les blocs-notes, les utilisateurs créent un ou plusieurs Workspaces dans un studio. Pour de plus amples informations, veuillez consulter Découvrir les bases de l'espace de Workspace.

À l'instar des Workspaces de JupyterLab, un Workspace préserve l'état du bloc-notes. Cependant, l'interface utilisateur de Workspace étend l'interface open source JupyterLab. Il offre des outils supplémentaires qui vous permettent de créer et de rattacher des clusters EMR, d'exécuter des tâches, d'explorer des exemples de blocs-notes et de lier des référentiels Git.

La liste suivante inclut les principales fonctionnalités des Workspaces EMR Studio :

- La visibilité du Workspace est basée sur le studio. Les Workspaces que vous créez dans un studio ne sont pas visibles dans les autres studios.
- Par défaut, un Workspace est partagé et peut être vu par tous les utilisateurs de Studio. Toutefois, un seul utilisateur peut ouvrir et travailler dans un Workspace à la fois. Pour travailler simultanément avec d'autres utilisateurs, vous pouvez Configuration de la collaboration dans Workspace
- Lorsque vous activez la collaboration dans un Workspace, vous pouvez collaborer simultanément avec d'autres utilisateurs dans un Workspace. Pour de plus amples informations, veuillez consulter Configuration de la collaboration dans Workspace.
- Les blocs-notes d'un Workspace partagent le même cluster EMR pour exécuter des commandes. Vous pouvez rattacher un Workspace à un cluster Amazon EMR exécuté sur Amazon EC2, ou à un cluster virtuel Amazon EMR sur EKS et à un point de terminaison géré.
- Les Workspaces peuvent basculer vers une autre zone de disponibilité que vous associez aux sous-réseaux d'un studio. Vous pouvez arrêter et redémarrer un Workspace pour lancer le processus de basculement. Lorsque vous redémarrez un Workspace et que le studio est configuré pour accéder à plusieurs zones de disponibilité, EMR Studio lance le Workspace dans une autre

zone de disponibilité au sein du VPC du studio. Si le studio ne possède qu'une seule zone de disponibilité, EMR Studio tente de lancer le Workspace dans un autre sous-réseau. Pour de plus amples informations, veuillez consulter Résoudre les problèmes de connectivité dans Workspace.

- Un Workspace peut se connecter à des clusters dans n'importe quel sous-réseau rattaché à un studio.

Pour plus d'informations sur la création et la configuration des Workspaces EMR Studio, consultez Découvrir les bases de l'espace de Workspace.

Stockage pour ordinateurs portables dans Amazon EMR Studio

Lorsque vous utilisez un Workspace, EMR Studio enregistre automatiquement les cellules des fichiers de bloc-notes à une cadence régulière dans l'emplacement Amazon S3 rattaché à votre studio. Ce processus de sauvegarde préserve le travail entre les sessions afin que vous puissiez y revenir ultérieurement sans avoir à apporter de modifications à un référentiel Git. Pour de plus amples informations, veuillez consulter Enregistrer le contenu Workspace.

Lorsque vous supprimez un fichier de bloc-notes d'un Workspace, EMR Studio supprime pour vous la version de sauvegarde d'Amazon S3. Toutefois, si vous supprimez un Workspace sans supprimer au préalable ses fichiers de bloc-notes, les fichiers du bloc-notes restent dans Amazon S3 et continuent d'entraîner des frais de stockage. Pour en savoir plus, consultez Supprimer un Workspace et des fichiers de bloc-notes.

Considérations relatives à EMR Studio

Considérations

Lorsque vous travaillez avec EMR Studio, tenez compte des facteurs suivants :

- EMR Studio est disponible dans les versions suivantes : Régions AWS
  - USA Est (Ohio) (us-east-2)
  - USA Est (Virginie du Nord) (us-east-1)
  - US Ouest (N. California) (us-west-1)
  - USA Ouest (Oregon) (us-west-2)
  - Afrique (Le Cap) (af-south-1)
  - Asie-Pacifique (Hong Kong) (ap-east-1)

- Asie-Pacifique (Jakarta) (ap-southeast-3) *
- Asie-Pacifique (Melbourne) (ap-southeast-4) *
- Asie-Pacifique (Mumbai) (ap-south-1)
- Asie-Pacifique (Osaka) (ap-northeast-3) *
- Asie-Pacifique (Séoul) (ap-northeast-2)
- Asie-Pacifique (Singapour) (ap-southeast-1)
- Asie-Pacifique (Sydney) (ap-southeast-2)
- Asie-Pacifique (Tokyo) (ap-northeast-1)
- Canada (Centre) (ca-central-1)
- Europe (Francfort) (eu-central-1)
- Europe (Irlande) (eu-west-1)
- Europe (Londres) (eu-west-2)
- Europe (Milan) (eu-south-1)
- Europe (Paris) (eu-west-3)
- Europe (Espagne) (eu-south-2)
- Europe (Stockholm) (eu-north-1)
- Europe (Zurich) (eu-central-2) *
- Israël (Tel Aviv) (il-central-1) *
- Moyen-Orient (Émirats arabes unis) (me-central-1) *
- Amérique du Sud (São Paulo) (sa-east-1)
- AWS GovCloud (USA Est) (gov-us-east-1)
- AWS GovCloud (US-Ouest) (gov-us-west-1)

  * L'interface utilisateur Live de Spark n'est pas prise en charge dans ces régions.

- Pour permettre aux utilisateurs de provisionner de nouveaux clusters EMR exécutés sur Amazon EC2 avec un Workspace, vous pouvez rattacher un EMR Studio à un ensemble de modèles de clusters. Les administrateurs peuvent définir des modèles de clusters avec Service Catalog et choisir si un utilisateur ou un groupe peut accéder aux modèles de clusters ou non dans un studio.
- Lorsque vous définissez des autorisations d'accès aux fichiers de bloc-notes stockés dans Amazon S3 ou que vous en lisez des secrets AWS Secrets Manager, utilisez le rôle de service Amazon EMR. Les politiques de session ne sont pas prises en charge avec ces autorisations.

- Vous pouvez créer plusieurs EMR Studio pour contrôler l'accès aux clusters EMR dans différents VPC.
- Utilisez le AWS CLI pour configurer Amazon EMR sur des clusters EKS. Vous pouvez ensuite utiliser l'interface Studio pour rattacher des clusters à des Workspaces avec un point de terminaison géré afin d'exécuter des tâches liées aux blocs-notes.
- D'autres considérations s'appliquent à EMR Studio lorsque vous utilisez la propagation d'identité approuvée avec Amazon EMR. Pour plus d'informations, consultez Considérations et limitations relatives à l'intégration d'Amazon EMR à Identity Center.
- EMR Studio ne prend pas en charge les commandes magiques suivantes en Python :
  - `%alias`
  - `%alias_magic`
  - `%automagic`
  - `%macro`
  - `%%js`
  - `%%javascript`
  - Modification de `proxy_user` à l'aide de `%configure`
  - Modification de `KERNEL_USERNAME` à l'aide de `%env` ou `%set_env`
- Amazon EMR sur les clusters EKS ne prend pas en charge SparkMagic les commandes pour EMR Studio.
- Pour écrire des instructions Scala multilignes dans des cellules du bloc-notes, assurez-vous que toutes les lignes, sauf la dernière, se terminent par un point. L'exemple suivant utilise la syntaxe correcte pour les instructions Scala multilignes.

```
val df = spark.sql("SELECT * from table_name).
        filter("col1=='value'").
        limit(50)
```

- Pour renforcer la sécurité des applications hors console que vous pouvez utiliser avec Amazon EMR, les domaines hébergeant les applications sont enregistrés dans la liste des suffixes publics (PSL). Voici des exemples de ces domaines d'hébergement : `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Pour plus de sécurité, si vous avez besoin de définir des cookies sensibles dans le nom de domaine par défaut, nous vous recommandons d'utiliser des cookies avec un préfixe `__Host-`. Cela vous permettra de protéger votre domaine contre les

tentatives de falsification de requêtes intersites (CSRF). Pour plus d'informations, voir la page Set-Cookie du Mozilla Developer Network.

Problèmes connus

- Un studio EMR qui utilise IAM Identity Center avec la propagation d'identité approuvée ne peut être associé qu'aux clusters EMR qui utilisent également la propagation d'identité approuvée.
- Avant de créer un Studio, assurez-vous de désactiver les outils de gestion de proxy comme FoxyProxy ou SwitchyOmega dans le navigateur. Les proxys actifs peuvent provoquer des erreurs lorsque vous choisissez Créer un studio et générer un message d'erreur de défaillance du réseau.
- Les noyaux qui s'exécutent sur Amazon EMR sur des clusters EKS peuvent ne pas démarrer en raison de problèmes d'expiration du délai. Si vous rencontrez une erreur ou un problème lors du démarrage du noyau, fermez le fichier de bloc-notes, arrêtez le noyau, puis rouvrez le fichier de bloc-notes.
- L'opération de redémarrage du noyau ne fonctionne pas comme prévu lorsque vous utilisez un cluster Amazon EMR sur EKS. Après avoir sélectionné Redémarrer le noyau, actualisez le Workspace pour que le redémarrage prenne effet.
- Si aucun Workspace n'est rattaché à un cluster, un message d'erreur s'affiche lorsqu'un utilisateur de Studio ouvre un fichier de bloc-notes et tente de sélectionner un noyau. Vous pouvez ignorer ce message d'erreur en choisissant Ok, mais vous devez rattacher le Workspace à un cluster et sélectionner un noyau avant de pouvoir exécuter le code du bloc-notes.
- Lorsque vous utilisez Amazon EMR 6.2.0 avec une configuration de sécurité pour configurer la sécurité du cluster, l'interface Workspace apparaît vide et ne fonctionne pas comme prévu. Si vous souhaitez configurer le chiffrement des données ou l'autorisation Amazon S3 pour EMRFS avec un cluster, nous vous recommandons d'utiliser une autre version prise en charge d'Amazon EMR. EMR Studio fonctionne avec les versions 5.32.0 (série Amazon EMR 5.x) ou 6.2.0 (série Amazon EMR 6.x) et les versions ultérieures d'Amazon EMR.
- Lorsque vous Déboguer Amazon EMR en cours d'exécution sur des tâches Amazon EC2, les liens vers l'interface utilisateur Spark intégrée au cluster peuvent ne pas fonctionner ou ne pas s'afficher. Pour régénérer les liens, créez une nouvelle cellule de bloc-notes et exécutez la commande `%%info`.
- Jupyter Enterprise Gateway ne nettoie pas les noyaux inactifs sur le nœud primaire d'un cluster dans les versions Amazon EMR suivantes : 5.32.0, 5.33.0, 6.2.0 et 6.3.0. Les noyaux inactifs consomment des ressources informatiques et peuvent entraîner la défaillance de clusters qui fonctionnent depuis longtemps. Vous pouvez configurer le nettoyage du noyau inactif pour

Jupyter Enterprise Gateway à l'aide de l'exemple de script suivant. Vous pouvez Connexion au nœud primaire à l'aide de SSH, ou soumettre le script en tant qu'étape. Pour plus d'informations, consultez Exécuter des commandes et des scripts sur un cluster Amazon EMR.

```
#!/bin/bash
sudo tee -a /emr/notebook-env/conf/jupyter_enterprise_gateway_config.py << EOF
c.MappingKernelManager.cull_connected = True
c.MappingKernelManager.cull_idle_timeout = 10800
c.MappingKernelManager.cull_interval = 300
EOF
sudo systemctl daemon-reload
sudo systemctl restart jupyter_enterprise_gateway
```

- Lorsque vous utilisez une politique d'arrêt automatique avec les versions 5.32.0, 5.33.0, 6.2.0 ou 6.3.0 d'Amazon EMR, Amazon EMR marque un cluster comme étant inactif et peut automatiquement le mettre fin à celui-ci, même si vous avez un noyau Python3 actif. Cela est dû au fait que l'exécution d'un noyau Python3 ne soumet pas de tâche Spark sur le cluster. Pour utiliser l'arrêt automatique avec un noyau Python3, nous vous recommandons d'utiliser Amazon EMR version 6.4.0 ou ultérieure. Pour plus d'informations sur l'arrêt automatique, consultez Utilisation d'une politique de résiliation automatique.
- Lorsque vous `%%display` affichez un Spark DataFrame dans un tableau, les tableaux très larges peuvent être tronqués. Cliquez avec le bouton droit sur la sortie et sélectionnez Créer une nouvelle vue pour la sortie afin d'obtenir une vue défilante de la sortie.
- Le démarrage d'un noyau basé sur Spark, tel que PySpark Spark ou SparkR, démarre une session Spark, et l'exécution d'une cellule dans un bloc-notes place les tâches Spark dans la file d'attente de cette session. Lorsque vous interrompez une cellule en cours d'exécution, la tâche Spark continue de s'exécuter. Pour arrêter la tâche Spark, vous devez utiliser l'interface utilisateur Spark intégrée au cluster. Pour plus d'informations sur la façon de se connecter à l'interface utilisateur Spark, consultez Déboguer des applications et des tâches avec EMR Studio.

Limites fonctionnelles

Amazon EMR Studio ne prend pas en charge les fonctionnalités Amazon EMR suivantes :

- Attacher et exécuter des tâches sur des clusters EMR avec une configuration de sécurité qui spécifie l'authentification Kerberos
- Clusters dotés de plusieurs nœuds primaires

- Clusters utilisant des instances Amazon EC2 basées sur AWS Graviton2 pour les versions 6.x d'Amazon EMR 6.x inférieures à la version 6.9.0 et les versions 5.x inférieures à la version 5.36.1

Les fonctionnalités suivantes ne sont pas prises en charge par un studio qui utilise la propagation d’identité approuvée :

- Création de clusters EMR sans modèle
- Utilisation d’applications EMR sans serveur
- Lancement d’Amazon EMR sur des clusters EKS
- Utilisation d’un rôle d’exécution
- Activation de la collaboration avec SQL Explorer ou Workspace

Limites de service pour EMR Studio

Le tableau suivant indique les limites de service pour EMR Studio.

Élément	Limite
EMR Studios	Maximum de 100 par AWS compte
Sous-réseaux	Maximum de 5 rattachés à chaque EMR Studio
Groupes IAM Identity Center	Maximum de 5 rattachés à chaque EMR Studio
Utilisateurs IAM Identity Center	Maximum de 100 rattachés à chaque EMR Studio

Bonnes pratiques en matière de VPC et de sous-réseaux

Suivez les meilleures pratiques suivantes pour configurer un Amazon Virtual Private Cloud (Amazon VPC) avec des sous-réseaux pour EMR Studio :

- Vous pouvez spécifier un maximum de cinq sous-réseaux dans votre VPC à rattacher au studio. Afin de garantir la disponibilité du Workspace et de permettre aux utilisateurs de Studio d'accéder à des clusters dans différentes zones de disponibilité, nous vous recommandons de fournir plusieurs sous-réseaux dans différentes zones de disponibilité. Pour en savoir plus sur l'utilisation des VPC,

des sous-réseaux et des zones de disponibilité, consultez la section VPC et sous-réseaux du Guide de l'utilisateur Amazon Virtual Private Cloud .

- Les sous-réseaux que vous spécifiez doivent pouvoir communiquer entre eux.
- Pour permettre aux utilisateurs de lier un Workspace à des référentiels Git hébergés sur un serveur public, vous devez spécifier uniquement les sous-réseaux privés ayant accès à Internet via la traduction d'adresses réseau (NAT). Pour plus d'informations sur la configuration d'un sous-réseau privé pour Amazon EMR, consultez Sous-réseaux privés.
- Lorsque vous utilisez Amazon EMR sur EKS avec EMR Studio, il doit y avoir au moins un sous-réseau commun entre votre studio et le cluster Amazon EKS que vous utilisez pour enregistrer un cluster virtuel. Dans le cas contraire, votre point de terminaison géré n'apparaîtra pas en tant qu'option dans les Workspaces Studio. Vous pouvez créer un cluster Amazon EKS et l'rattacher à un sous-réseau appartenant au studio, ou créer un studio et spécifier les sous-réseaux de votre cluster EKS.
- Si vous prévoyez d'utiliser Amazon EMR sur EKS avec EMR Studio, choisissez le même VPC que vos composants master de cluster Amazon EKS.

Exigences relatives au cluster pour Amazon EMR Studio

Clusters Amazon EMR exécutés sur Amazon EC2

Tous les clusters Amazon EMR exécutés sur Amazon EC2 que vous créez pour un Workspace EMR Studio doivent répondre aux exigences suivantes. Les clusters que vous créez à l'aide de l'interface EMR Studio répondent automatiquement à ces exigences.

- Le cluster doit fonctionner avec les versions 5.32.0 (série Amazon EMR 5.x) ou 6.2.0 (série Amazon EMR 6.x) et versions ultérieures d'Amazon EMR. Vous pouvez créer un cluster à l'aide de la console Amazon EMR, ou SDK AWS Command Line Interface, puis l'associer à un espace de travail EMR Studio. Les utilisateurs de Studio peuvent également provisionner et rattacher des clusters lors de la création ou de l'utilisation d'un Workspace Amazon EMR. Pour plus d’informations, consultez Attacher un calcul à un espace de travail EMR Studio.
- Le cluster doit se trouver dans un cloud privé virtuel Amazon. La plateforme EC2-Classic n'est pas prise en charge.
- Spark, Livy et Jupyter Enterprise Gateway doivent être installés sur le cluster. Si vous envisagez d'utiliser le cluster pour SQL Explorer, vous devez installer Presto et Spark.
- Pour utiliser SQL Explorer, le cluster doit fonctionner avec Amazon EMR version 5.34.0 ou ultérieure ou version 6.4.0 ou ultérieure, et Presto doit être installé. Si vous souhaitez spécifier le

catalogue AWS Glue Data comme métastore Hive pour Presto, vous devez le configurer sur le cluster. Pour plus d'informations, consultez Utilisation de Presto avec le catalogue de données AWS Glue.

- Le cluster doit se trouver dans un sous-réseau privé avec traduction d'adresses réseau (NAT) pour utiliser les référentiels Git hébergés publiquement avec EMR Studio.

Lorsque vous travaillez avec EMR Studio, nous recommandons les configurations de cluster suivantes.

- Définissez le mode de déploiement pour les sessions Spark sur le mode Cluster. Le mode Cluster place les processus principaux de l'application sur les nœuds principaux et non sur le nœud primaire d'un cluster. Cela soulage le nœud primaire des pressions potentielles sur la mémoire. Pour plus d'informations, consultez Présentation du mode cluster dans la documentation Apache Spark.
- Modifiez le délai d'expiration de Livy d'une heure par défaut à six heures, comme dans l'exemple de configuration suivant.

```
{
    "classification":"livy-conf",
        "Properties":{
            "livy.server.session.timeout":"6h",
            "livy.spark.deploy-mode":"cluster"
        }
}
```

- Créez des flottes d'instances variées comprenant jusqu'à 30 instances et sélectionnez plusieurs types d'instances dans votre flotte d'instances Spot. Par exemple, pour les charges de travail Spark, vous pouvez spécifier les types d'instances à mémoire optimisée suivants : r5.2x, r5.4x, r5.8x, r5.12x, r5.16x, r4.2x, r4.4x, r4.8x, r4.12, etc. Pour plus d'informations, consultez Configuration de parcs d'instances.
- Utilisez la stratégie d'allocation optimisée pour les instances Spot afin d'aider Amazon EMR à effectuer des sélections d'instances efficaces, sur la base des informations de capacité en temps réel fournies par Amazon EC2. Pour plus d'informations, consultez Stratégie d'allocation pour les flottes d'instance.
- Activer la mise à l'échelle gérée sur votre cluster. Définissez le paramètre « Nombre maximal de nœuds principaux » sur la capacité persistante minimale que vous prévoyez d'utiliser, puis, afin de réduire les coûts, configurez la mise à l'échelle sur une flotte de tâches bien diversifiée qui

s'exécute sur des instances Spot. Pour plus d'informations, consultez Utiliser la mise à l'échelle gérée dans Amazon EMR.

Nous vous conseillons également de laisser Amazon EMR Block Public Access activé, afin de limiter le trafic SSH entrant à des sources fiables. L'accès entrant à un cluster permet aux utilisateurs d'exécuter des blocs-notes sur le cluster. Pour plus d'informations, consultez Utilisation du blocage de l'accès public Amazon EMR et Contrôle du trafic réseau avec des groupes de sécurité.

Amazon EMR sur des clusters EKS

Outre les clusters EMR exécutés sur Amazon EC2, vous pouvez configurer et gérer des clusters Amazon EMR sur EKS pour EMR Studio à l'aide de l'interface AWS CLI. Configurez Amazon EMR sur des clusters EKS en suivant les directives suivantes :

- Créez un point de terminaison HTTPS géré pour le cluster Amazon EMR sur EKS. Les utilisateurs associent un Workspace à un point de terminaison géré. Le cluster Amazon Elastic Kubernetes Service (EKS) que vous utilisez pour enregistrer un cluster virtuel doit disposer d'un sous-réseau privé pour prendre en charge les points de terminaison gérés.
- Utilisez un cluster Amazon EKS avec au moins un sous-réseau privé et une traduction d'adresses réseau (NAT) lorsque vous souhaitez utiliser des référentiels Git hébergés sur un serveur public.
- Évitez d'utiliser les AMI Arm Amazon Linux optimisées par Amazon EKS : elles ne sont pas prises en charge pour Amazon EMR sur les points de terminaison gérés par EKS.
- Évitez d'utiliser AWS Fargate uniquement des clusters Amazon EKS, qui ne sont pas pris en charge.

Configurer Amazon EMR Studio

Cette section est destinée aux administrateurs EMR Studio. Elle explique comment configurer un EMR Studio pour votre équipe et fournit des instructions pour des tâches telles que l'attribution d'utilisateurs et de groupes, la configuration de modèles de clusters et l'optimisation d'Apache Spark pour EMR Studio.

Rubriques

- Autorisations d'administrateur pour créer et gérer un EMR Studio
- Configurer un Amazon EMR Studio
- Gérer un Amazon EMR Studio

- Chiffrement des blocs-notes et des fichiers de l'espace de travail EMR Studio
- Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio
- Créer des modèles AWS CloudFormation pour Amazon EMR Studio
- Établissez l'accès et les autorisations pour les référentiels Git
- Optimiser les tâches Spark dans EMR Studio

Autorisations d'administrateur pour créer et gérer un EMR Studio

Les autorisations IAM décrites sur cette page vous permettent de créer et de gérer un EMR Studio. Pour plus d'informations sur chaque autorisation requise, consultez Autorisations requises pour gérer un EMR Studio.

Autorisations requises pour gérer un EMR Studio

Le tableau suivant répertorie les opérations liées à la création et à la gestion d'un EMR Studio. Le tableau affiche également les autorisations nécessaires pour chaque opération.

> **Note**
>
> Vous n'avez besoin des actions `SessionMapping` IAM Identity Center et Studio que lorsque vous utilisez le mode d'authentification IAM Identity Center.

Autorisations pour créer et gérer un EMR Studio

Opération	Autorisations
Créer un Studio	`"elasticmapreduce:CreateStudio", "sso:CreateApplication", "sso:PutApplicationAuthenticationMethod", "sso:PutApplicationGrant", "sso:PutApplicationAccessScope", "sso:PutApplicationAssignmentConfiguration", "iam:PassRole"`
Décrire un Studio	`"elasticmapreduce:DescribeStudio",`

Opération	Autorisations
	`"sso:GetManagedApplicationInstance"`
Répertorier des Studios	`"elasticmapreduce:ListStudios"`
Supprimer un Studio	`"elasticmapreduce:DeleteStudio", "sso:DeleteApplication", "sso:DeleteApplicationAuthentica tionMethod", "sso:DeleteApplicationAccessScope", "sso:DeleteApplicationGrant"`
Additional permissions required when you use IAM Identity Center mode	
Attribuer des utilisateurs ou des groupes à un Studio	`"elasticmapreduce:CreateStudioSessio nMapping", "sso:GetProfile", "sso:ListDirectoryAssociations", "sso:ListProfiles", "sso:AssociateProfile", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListInstances", "sso:CreateApplicationAssignment", "sso:DescribeInstance", "organizations:DescribeOrga nization", "organizations:ListDelegatedAdmini strators", "sso:CreateInstance", "sso:DescribeRegisteredRegions", "sso:GetSharedSsoConfiguration", "iam:ListPolicies"`

Opération	Autorisations
Récupérer les détails des attributions Studio pour un utilisateur ou un groupe spécifique	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "elasticmapreduce:GetStudioSessionMapping"`
Répertorier tous les utilisateurs et groupes attribués à un Studio	`"elasticmapreduce:ListStudioSessionMappings"`
Mettre à jour la politique de session attachée à un utilisateur ou à un groupe attribué à un Studio	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "sso:DescribeInstance", "elasticmapreduce:UpdateStudioSessionMapping"`
Supprimer un utilisateur ou un groupe d'un Studio	`"elasticmapreduce:DeleteStudioSessionMapping", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListDirectoryAssociations", "sso:GetProfile", "sso:DescribeApplication", "sso:DescribeInstance", "sso:ListProfiles", "sso:DisassociateProfile", "sso:DeleteApplicationAssignment", "sso:ListApplicationAssignments"`

Pour créer une politique avec des autorisations d'administrateur pour EMR Studio

1. Suivez les instructions de la rubrique Création de politiques IAM pour créer une politique à l'aide de l'un des exemples suivants. Les autorisations dont vous avez besoin dépendent de votre mode d'authentification pour EMR Studio.

   Saisissez vos propres valeurs pour ces éléments :

   - Remplacez *<your-resource-ARN>* pour spécifier l'Amazon Resource Name (ARN) de l'objet ou des objets couverts par la déclaration pour vos cas d'utilisation.
   - Remplacez *<region>* par le code de l'Région AWS où vous souhaitez créer le Studio.
   - Remplacez *<aws-account_id>* par l'ID du compte AWS pour le Studio.
   - Remplacez *<EMRStudio-Service-Role>* et *<EMRStudio-User-Role>* par les noms de votre fonction du service EMR Studio et de votre rôle d'utilisateur EMR Studio.

   Example Exemple de politique : autorisations d'administrateur lorsque vous utilisez le mode d'authentification IAM

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
            "Action": [
                "elasticmapreduce:ListStudios"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>"
            ],
            "Action": "iam:PassRole"
        }
    ]
}
```

Example Exemple de politique : autorisations d'administrateur lorsque vous utilisez le mode d'authentification IAM Identity Center

> **Note**
>
> Les API d'Identity Center et du répertoire associé ne prennent pas en charge la spécification d'un ARN dans l'élément de ressource d'une déclaration de politique IAM. Pour autoriser l'accès à IAM Identity Center et à IAM Identity Center Directory, les autorisations suivantes spécifient toutes les ressources, "Resource":"*", pour les actions IAM Identity Center. Pour de plus amples informations, veuillez consulter Actions, ressources et clés de condition pour IAM Identity Center Directory.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:CreateStudioSessionMapping",
                "elasticmapreduce:GetStudioSessionMapping",
                "elasticmapreduce:UpdateStudioSessionMapping",
                "elasticmapreduce:DeleteStudioSessionMapping"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
```

```
            "Action": [
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:ListStudioSessionMappings"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>",
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-User-Role>"
            ],
            "Action": "iam:PassRole"
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "sso:CreateApplication",
                "sso:PutApplicationAuthenticationMethod",
                "sso:PutApplicationGrant",
                "sso:PutApplicationAccessScope",
                "sso:PutApplicationAssignmentConfiguration",
                "sso:DescribeApplication",
                "sso:DeleteApplication",
                "sso:DeleteApplicationAuthenticationMethod",
                "sso:DeleteApplicationAccessScope",
                "sso:DeleteApplicationGrant",
                "sso:ListInstances",
                "sso:CreateApplicationAssignment",
                "sso:DeleteApplicationAssignment",
                "sso:ListApplicationAssignments",
                "sso:DescribeInstance",
                "sso:AssociateProfile",
                "sso:DisassociateProfile",
                "sso:GetProfile",
                "sso:ListDirectoryAssociations",
                "sso:ListProfiles",
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
                "sso:CreateInstance",
```

```
                "sso:DescribeRegisteredRegions",
                "sso:GetSharedSsoConfiguration",
                "iam:ListPolicies"
            ]
        }
    ]
}
```

2. Attachez la politique à votre identité IAM (groupe, rôle ou groupe). Pour de plus amples instructions, consultez la rubrique Ajout et suppression d'autorisations basées sur l'identité IAM.

Configurer un Amazon EMR Studio

Procédez comme suit pour configurer un Amazon EMR Studio.

Avant de commencer

> **Note**
>
> Si vous envisagez d'utiliser EMR Studio avec Amazon EMR sur EKS, nous vous recommandons de configurer Amazon EMR sur EKS pour EMR Studio avant de configurer un Studio.

Avant de configurer un EMR Studio, assurez-vous d'avoir les éléments suivants :

- Un Compte AWS. Pour obtenir des instructions, veuillez consulter Configuration d'Amazon EMR.
- Les autorisations pour créer et gérer un EMR Studio. Pour plus d'informations, consultez the section called "Autorisations d'administrateur pour créer un EMR Studio".
- Un compartiment Amazon S3 dans lequel EMR Studio peut sauvegarder les espaces de travail et les fichiers de bloc-notes de votre Studio. Pour obtenir des instructions, consultez la rubrique Créer un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service (S3).
- Si vous souhaitez vous connecter à un cluster Amazon EMR sur EC2 ou Amazon EMR sur EKS, ou utiliser des référentiels Git, vous avez besoin d'un cloud privé virtuel (VPC) Amazon pour le Studio et d'un maximum de cinq sous-réseaux. Vous n'avez pas besoin d'un VPC pour utiliser EMR Studio avec EMR sans serveur. Pour obtenir des conseils sur la configuration réseaux, consultez Bonnes pratiques en matière de VPC et de sous-réseaux.

Pour configurer un EMR Studio

1. Choisir un mode d'authentification pour Amazon EMR Studio
2. Créez les ressources Studio suivantes.
   - Créer une fonction du service EMR Studio
   - Configurer les autorisations utilisateur d'EMR Studio pour Amazon EC2 ou Amazon EKS
   - (Facultatif) Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio.
3. Créer un EMR Studio
4. Attribuer un utilisateur ou un groupe à un EMR Studio

Une fois que vous avez terminé les étapes de configuration, vous pouvez Utiliser un Amazon EMR Studio.

Choisir un mode d'authentification pour Amazon EMR Studio

EMR Studio prend en charge deux modes d'authentification : le mode d'authentification IAM et le mode d'authentification IAM Identity Center. Le mode IAM utilise AWS Identity and Access Management (IAM), tandis que le mode IAM Identity Center utilise AWS IAM Identity Center. Lorsque vous créez un EMR Studio, vous choisissez le mode d'authentification pour tous les utilisateurs de ce studio. Pour plus d'informations sur mes différents modes d'authentification, consultez Authentification et connexion utilisateur.

Utilisez le tableau suivant pour choisir un mode d'authentification pour EMR Studio.

Si...	Il est recommandé de...
Vous connaissez déjà ou avez déjà configuré l'authentification ou la fédération IAM	Mode d'authentification IAM, qui offre les avantages suivants : • permet de configurer rapidement EMR Studio si vous gérez déjà des identités telles que des utilisateurs et des groupes dans IAM ; • fonctionne avec les fournisseurs d'identité qui sont compatibles avec OpenID Connect (OIDC) ou Security Assertion Markup Language 2.0 (SAML 2.0) ;

Si...	Il est recommandé de...
	• prend en charge l'utilisation de plusieurs fournisseurs d'identité avec le même Compte AWS ; • disponible dans un grand nombre d'Régions AWS ; • conforme à la norme SOC 2.
Vous ne connaissez pas AWS ou Amazon EMR	Mode d'authentification IAM Identity Center, qui offre les fonctionnalités suivantes : • facilite l'attribution des ressources AWS aux utilisateurs et aux groupes ; • fonctionne avec les fournisseurs d'identité Microsoft Active Directory et SAML 2.0 ; • Facilite la configuration de la fédération à comptes multiples afin que vous n'ayez pas à configurer la fédération séparément pour chaque Compte AWS de votre organisation.

Configurer le mode d'authentification IAM pour Amazon EMR Studio

Avec le mode d'authentification IAM, vous pouvez utiliser l'authentification IAM ou la fédération IAM. L'authentification IAM vous permet de gérer les identités IAM telles que les utilisateurs, les groupes et les rôles dans IAM. Vous autorisez les utilisateurs à accéder à un studio avec des politiques d'autorisations IAM et un contrôle d'accès par attributs (ABAC). La fédération IAM vous permet d'établir un lien de confiance entre un fournisseur d'identité (IdP) tiers et AWS afin que vous puissiez gérer les identités des utilisateurs par le biais de votre IdP.

> **Note**
>
> Si vous utilisez déjà IAM pour contrôler l'accès aux ressources AWS, ou si vous avez déjà configuré votre fournisseur d'identité (IdP) pour IAM, consultez Autorisations utilisateur pour le mode d'authentification IAM pour définir les autorisations utilisateur lorsque vous utilisez le mode d'authentification IAM pour EMR Studio.

Utiliser la fédération IAM pour Amazon EMR Studio

Pour utiliser la fédération IAM pour EMR Studio, vous devez créer une relation de confiance entre votre Compte AWS et votre fournisseur d'identité (IdP) et permettre aux utilisateurs fédérés d'accéder à l'AWS Management Console. Les étapes à suivre pour créer cette relation de confiance varient en fonction de la norme de fédération de votre IdP.

En général, vous devez effectuer les tâches suivantes pour configurer la fédération avec un IdP externe. Pour obtenir des instructions complètes, consultez les rubriques Activation de l'accès des utilisateurs fédérés SAML 2.0 à la AWS Management Console et Activation de l'accès de broker d'identité personnalisé à la AWS Management Console dans le Guide de l'utilisateur AWS Identity and Access Management.

1. Récoltez des informations sur votre IdP. Cela implique généralement de générer un document de métadonnées pour valider les requêtes d'authentification SAML de votre IdP.
2. Créez une entité IAM de fournisseur d'identité pour stocker les informations sur votre IdP. Pour obtenir des instructions, consultez la rubrique Création de fournisseurs d'identité IAM.
3. Créez un ou plusieurs rôles IAM pour votre IdP. EMR Studio attribue un rôle à un utilisateur fédéré lorsque ce dernier se connecte. Le rôle permet à votre IdP de demander des informations d'identification de sécurité temporaires pour accéder à AWS. Pour obtenir des instructions, consultez la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération). Les politiques d'autorisation que vous attribuez au rôle déterminent ce que les utilisateurs fédérés peuvent faire dans AWS et dans un EMR Studio. Pour plus d'informations, consultez Autorisations utilisateur pour le mode d'authentification IAM.
4. (Pour les fournisseurs SAML) Complétez la relation d'approbation SAML en configurant votre IdP à l'aide d'informations sur AWS et les rôles que vous voulez attribuer à des utilisateurs fédérés. Ce processus de configuration crée une relation de confiance entre votre IdP et AWS. Pour plus d'informations, consultez Configuration de votre IdP SAML 2.0 à l'aide d'une relation d'approbation des parties utilisatrices et ajout de demandes.

Pour configurer un EMR Studio en tant qu'application SAML sur votre portail IdP

Vous pouvez configurer un EMR Studio spécifique en tant qu'application SAML à l'aide d'un lien ciblé vers le Studio. Cela permet aux utilisateurs de se connecter à votre portail IdP et de lancer un Studio spécifique au lieu de naviguer via la console Amazon EMR.

- Utilisez le format suivant pour configurer un lien ciblé vers votre EMR Studio en tant qu'URL de destination après vérification des assertions SAML.

```
https://console.aws.amazon.com/emr/home?region=<aws-region>#studio/<your-studio-id>/start
```

Configurer le mode d'authentification IAM Identity Center pour Amazon EMR Studio

Pour préparer AWS IAM Identity Center pour EMR Studio, vous devez configurer votre source d'identité et allouer les utilisateurs et les groupes. L'allocation est le processus qui consiste à mettre les informations des utilisateurs et des groupes à la disposition d'IAM Identity Center et des applications qui utilisent IAM Identity Center. Pour plus d'informations, consultez Provisionnement d'utilisateurs et de groupes.

EMR Studio prend en charge l'utilisation des fournisseurs d'identité suivants pour IAM Identity Center :

- AWS Managed Microsoft AD et Active Directory autogéré : pour plus d'informations, consultez Se connecter à votre annuaire Microsoft AD.
- Fournisseurs basés sur le protocole SAML : pour une liste complète, consultez Fournisseurs d'identité pris en charge.
- Le répertoire IAM Identity Center : pour plus d'informations, consultez Gérer les identités dans IAM Identity Center.

Pour configurer IAM Identity Center pour EMR Studio

1. Pour configurer IAM Identity Center pour EMR Studio, vous avez besoin des éléments suivants :
   - Un compte de gestion dans votre organisation AWS si vous utilisez plusieurs comptes dans votre organisation.

     > **Note**
     >
     > Vous ne devez utiliser votre compte de gestion que pour activer IAM Identity Center et allouer des utilisateurs et des groupes. Après avoir configuré IAM Identity Center, utilisez un compte membre pour créer un EMR Studio et attribuer des utilisateurs et des groupes. Pour en savoir plus sur la terminologie AWS, consultez Terminologie et concepts relatifs à AWS Organizations.

- Si vous avez activé IAM Identity Center avant le 25 novembre 2019, vous devrez peut-être activer les applications qui utilisent IAM Identity Center pour les comptes de votre organisation AWS. Pour plus d'informations, consultez Enable IAM Identity Center-integrated applications in AWS accounts.
- Assurez-vous de remplir les prérequis sur la page des prérequis pour IAM Identity Center.

2. Suivez les instructions de la rubrique Enable IAM Identity Center pour activer IAM Identity Center dans l'Région AWS où vous souhaitez créer l'EMR Studio.
3. Connectez IAM Identity Center à votre fournisseur d'identité et configurez les utilisateurs et les groupes que vous souhaitez attribuer au Studio.

Si vous utilisez...	Faites ceci...
Un annuaire Microsoft AD	1. Suivez les instructions de la rubrique Connect to your Microsoft AD directory pour connecter votre Active Directory ou votre annuaire AWS Managed Microsoft AD autogéré à l'aide d'AWS Directory Service. 2. Pour allouer des utilisateurs et des groupes pour IAM Identity Center, vous pouvez synchroniser les données d'identité de votre AD source pour IAM Identity Center. Vous pouvez synchroniser les identités à partir de votre AD source de nombreuses manières. L'une des méthodes consiste à attribuer des utilisateurs ou des groupes AD à un compte AWS au sein de votre organisation. Pour obtenir des instructions, consultez la rubrique relative à l'authentification unique. La synchronisation peut prendre jusqu'à deux heures. Une fois cette étape accomplie, les utilisateurs et groupes synchronisés apparaissent dans votre Identity Store.

Si vous utilisez...	Faites ceci...
	Note Les utilisateurs et les groupes n'apparaissent pas dans votre magasin d'identités tant que vous n'avez pas synchronisé les informations relatives aux utilisateurs et aux groupes ou que vous n'avez pas utilisé le provisionnement utilisateur just-in-time (JIT). Pour plus d'informations, consultez la rubrique Provisionnement lorsque les utilisateurs proviennent d'Active Directory. 3. (Facultatif) Après avoir synchronisé les utilisateurs et les groupes AD, vous pouvez supprimer leur accès à votre compte AWS que vous avez configuré à l'étape précédente. Pour plus d'instructions, consultez Remove user access.
Un fournisseur d'identité externe	Suivez les instructions de la rubrique Connect to your external identity provider.
Le répertoire IAM Identity Center	Lorsque vous créez des utilisateurs et des groupes dans IAM Identity Center, l'allocation est automatique. Pour plus d'informations, consultez la rubrique Manage identities in IAM Identity Center.

Vous pouvez désormais attribuer des utilisateurs et des groupes de votre Identity Store à un EMR Studio. Pour obtenir des instructions, veuillez consulter Attribuer un utilisateur ou un groupe à un EMR Studio.

Créer une fonction du service EMR Studio

À propos de la fonction du service EMR Studio

Chaque EMR Studio utilise un rôle IAM doté d'autorisations lui permettant d'interagir avec d'autres services AWS. Cette fonction du service doit inclure des autorisations permettant à EMR Studio d'établir un canal réseau sécurisé entre les espaces de travail et les clusters, de stocker des fichiers de bloc-notes dans Amazon S3 Control et d'accéder à AWS Secrets Manager tout en liant un espace de travail à un référentiel Git.

Utilisez la fonction du service Studio (au lieu des politiques de session) pour définir toutes les autorisations d'accès Amazon S3 pour le stockage des fichiers de bloc-notes et pour définir les autorisations d'accès AWS Secrets Manager.

Comment créer une fonction du service pour EMR Studio sur Amazon EC2 ou Amazon EKS

1. Suivez les instructions de la rubrique Création d'un rôle pour la délégation d'autorisations à un service AWS pour créer la fonction du service en utilisant la politique d'approbation suivante.

   > **Important**
   >
   > La politique d'approbation suivante inclut les clés de condition globales `aws:SourceArn` et `aws:SourceAccount` pour limiter les autorisations que vous accordez à EMR Studio pour des ressources particulières de votre compte. Cela peut vous protéger contre le problème de l'adjoint confus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
```

```
        "ArnLike": {
          "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
        }
      }
    }
  ]
}
```

2. Supprimez les autorisations de rôle par défaut. Incluez ensuite les autorisations issues de l'exemple de politique d'autorisation IAM suivant. Vous pouvez également créer une politique personnalisée qui utilise les Autorisations relatives aux fonctions de service EMR Studio.

> **Important**
>
> - Pour que le contrôle d'accès Amazon EC2 basé sur les balises fonctionne avec EMR Studio, vous devez définir l'accès pour l'API `ModifyNetworkInterfaceAttribute` comme indiqué dans la politique suivante.
> - Pour qu'EMR Studio puisse fonctionner avec la fonction du service, les déclarations `AllowAddingEMRTagsDuringDefaultSecurityGroupCreation` et `AllowAddingTagsDuringEC2ENICreation` doivent rester inchangées.
> - Pour utiliser l'exemple de politique, vous devez baliser les ressources suivantes avec la clé **`"for-use-with-amazon-emr-managed-policies"`** et la valeur **`"true"`**.
>   - Votre cloud privé virtuel (VPC) Amazon pour EMR Studio.
>   - Chaque sous-réseau que vous voulez utiliser avec le Studio.
>   - Tous les groupes de sécurité EMR Studio personnalisés. Vous devez baliser tous les groupes de sécurité que vous avez créés pendant la période d'essai d'EMR Studio si vous souhaitez continuer à les utiliser.
>   - Secrets conservés dans AWS Secrets Manager que les utilisateurs Studio utilisent pour lier les référentiels Git à un espace de travail.
>
> Vous pouvez appliquer des balises aux ressources à l'aide de l'onglet Balises sur l'écran des ressources correspondant dans l'AWS Management Console.

Le cas échéant, remplacez `*` dans `"Resource":"*"` par la politique suivante pour spécifier l'Amazon Resource Name (ARN) des ressources concernées par l'instruction pour votre cas d'utilisation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2ENIActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENIAttributeAction",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifyNetworkInterfaceAttribute"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowEC2SecurityGroupActionsWithEMRTags",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:DeleteNetworkInterfacePermission"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
```

```
    },
    {
      "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
          "ec2:CreateAction": "CreateSecurityGroup"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationInSubnetAndSecurityGroupWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
```

```
      }
    },
    {
      "Sid": "AllowAddingTagsDuringEC2ENICreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },
    {
      "Sid": "AllowEC2ReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowWorkspaceCollaboration",
      "Effect": "Allow",
```

```
      "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

3. Donnez à votre fonction du service un accès en lecture et en écriture à votre emplacement Amazon S3 pour EMR Studio. Utilisez l'ensemble minimum d'autorisations suivant. Pour plus d'informations, consultez l'exemple Amazon S3 : autorise l'accès en lecture et en écriture aux objets d'un compartiment S3, par programmation et dans la console.

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

   Si vous chiffrez votre compartiment Amazon S3, incluez les autorisations suivantes pour AWS Key Management Service.

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

4. Pour contrôler l’accès aux secrets Git au niveau utilisateur, ajoutez des autorisations basées sur des balises à `secretsmanager:GetSecretValue` dans la politique de rôle d’utilisateur d’EMR Studio et supprimez les autorisations de la politique `secretsmanager:GetSecretValue` de la politique de la fonction du service EMR Studio. Pour plus d’informations sur la définition d’autorisations utilisateur précises, voir la rubrique Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio.

Fonction du service minimum pour EMR sans serveur

Si vous souhaitez exécuter des charges de travail interactives avec EMR sans serveur via des blocs-notes EMR Studio, appliquez la même politique d'approbation que celle que vous avez utilisée pour configurer EMR Studio dans la section précédente, Comment créer une fonction du service pour EMR Studio sur Amazon EC2 ou Amazon EKS.

Pour votre Politique IAM, la politique minimale viable comporte les autorisations suivantes. Mettez à jour *bucket-name* avec le nom du compartiment que vous prévoyez d'utiliser lors de la configuration de votre EMR Studio et de votre espace de travail. EMR Studio utilise le compartiment pour sauvegarder les espaces de travail et les fichiers de bloc-notes de votre Studio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ObjectActions",
            "Effect": "Allow",
            "Action": [
               "s3:PutObject",
               "s3:GetObject",
               "s3:DeleteObject"
            ],
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        },
        {
            "Sid": "BucketActions",
            "Effect": "Allow",
            "Action": [
               "s3:ListBucket",
               "s3:GetEncryptionConfiguration"
            ],
            "Resource": ["arn:aws:s3:::bucket-name"]
        }
    ]
}
```

Si vous prévoyez d'utiliser un compartiment Amazon S3 chiffré, ajoutez les autorisations suivantes à votre politique :

```
"kms:Decrypt",
"kms:GenerateDataKey",
```

```
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

Autorisations relatives aux fonctions de service EMR Studio

Le tableau suivant répertorie les opérations effectuées par EMR Studio à l'aide de la fonction du service, ainsi que les actions IAM requises pour chaque opération.

Opération	Actions

Établir un canal réseau sécurisé entre un espace de travail et un cluster EMR, et effectuer les actions de nettoyage nécessaires.	`"ec2:CreateNetworkInterface",`
	`"ec2:CreateNetworkInterfacePermission",`
	`"ec2:DeleteNetworkInterface",`
	`"ec2:DeleteNetworkInterfacePermission",`
	`"ec2:DescribeNetworkInterfaces",`
	`"ec2:ModifyNetworkInterfaceAttribute",`
	`"ec2:AuthorizeSecurityGroupEgress",`
	`"ec2:AuthorizeSecurityGroupIngress",`
	`"ec2:CreateSecurityGroup",`
	`"ec2:DescribeSecurityGroups",`
	`"ec2:RevokeSecurityGroupEgress",`
	`"ec2:DescribeTags",`
	`"ec2:DescribeInstances",`
	`"ec2:DescribeSubnets",`
	`"ec2:DescribeVpcs",`
	`"elasticmapreduce:ListInstances",`
	`"elasticmapreduce:DescribeCluster",`
	`"elasticmapreduce:ListSteps"`

Utiliser les informations d'identification Git stockées dans AWS Secrets Manager pour lier des référentiels Git à un espace de travail.	`"secretsmanager:GetSecretValue"`
Appliquer les balises AWS à l'interface réseau et aux groupes de sécurité par défaut qu'EMR Studio crée lors de la configuration du canal	`"ec2:CreateTags"`

Opération	Actions
réseau sécurisé. Pour plus d'informations, veuillez consulter la rubrique Balisage des ressources AWS.	

Accédez ou chargez les fichiers de bloc-notes et les métadonnées sur Amazon S3.	`"s3:PutObject",`
	`"s3:GetObject",`
	`"s3:GetEncryptionConfiguration",`
	`"s3:ListBucket",`
	`"s3:DeleteObject"`
	
	Si vous utilisez un compartiment Amazon S3 chiffré, incluez les autorisations suivantes.
	
	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`
	`"kms:ReEncryptFrom",`
	`"kms:ReEncryptTo",`
	`"kms:DescribeKey"`

Activer et configurer la collaboration dans l'espace de travail.	`"iam:GetUser",`
	`"iam:GetRole",`
	`"iam:ListUsers",`
	`"iam:ListRoles",`
	`"sso:GetManagedApplicationInstance",`
	`"sso-directory:SearchUsers"`

Chiffrez les blocs-notes et les fichiers de l'espace de travail EMR Studio à l'aide de clés gérées par le client (CMK) avec AWS Key Management Service	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`
	`"kms:ReEncryptFrom",`
	`"kms:ReEncryptTo",`
	`"kms:DescribeKey"`

Configurer les autorisations utilisateur d'EMR Studio pour Amazon EC2 ou Amazon EKS

Vous devez configurer les politiques d'autorisation utilisateur pour Amazon EMR Studio afin de pouvoir définir des autorisations détaillées pour les utilisateurs et les groupes. Pour plus d'informations sur le fonctionnement des autorisations utilisateur dans EMR Studio, consultez Contrôle d'accès dans Comment fonctionne Amazon EMR Studio.

> **Note**
>
> Les autorisations abordées dans cette section n'appliquent pas le contrôle d'accès aux données. Pour gérer l'accès aux jeux de données d'entrée, vous devez configurer les autorisations pour les clusters utilisés par votre Studio. Pour plus d'informations, consultez Sécurité dans Amazon EMR.

Créer un rôle d'utilisateur EMR Studio pour le mode d'authentification IAM Identity Center

Vous devez créer un rôle d'utilisateur EMR Studio lorsque vous utilisez le mode d'authentification IAM Identity Center.

Pour créer un rôle d'utilisateur pour EMR Studio

1. Suivez les instructions de la rubrique Création d'un rôle pour la délégation d'autorisations à un service AWS dans le Guide de l'utilisateur AWS Identity and Access Management pour créer un rôle d'utilisateur.

   Utilisez la politique de relation d'approbation suivante lorsque vous créez le rôle.

   ```
   {
     "Version": "2008-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "elasticmapreduce.amazonaws.com"
         },
         "Action": [
                 "sts:AssumeRole",
                 "sts:SetContext"
                 ]
   ```

```
    }
  ]
}
```

2. Supprimez les autorisations et les politiques de rôle par défaut.
3. Associez vos politiques de session EMR Studio au rôle d'utilisateur avant d'affecter des utilisateurs et des groupes à un Studio. Pour obtenir des instructions sur la création de politiques de session, consultez Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio.

Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio

Reportez-vous aux sections suivantes pour créer des politiques d'autorisations pour EMR Studio.

Rubriques

- Création de politiques d'autorisations
- Définir la propriété pour la collaboration dans l'espace de travail
- Création d'une politique de secrets Git de niveau utilisateur
- Associez les politiques d'autorisations à votre identité IAM.

> **Note**
>
> Utilisez la fonction du service EMR Studio pour définir les autorisations d'accès Amazon S3 pour le stockage des fichiers de bloc-notes et les autorisations d'accès AWS Secrets Manager pour lire les secrets lors de la liaison des Workspaces aux référentiels Git.

Création de politiques d'autorisations

Créez une ou plusieurs politiques d'autorisations IAM qui spécifient les actions qu'un utilisateur peut effectuer dans votre Studio. Par exemple, vous pouvez créer trois politiques distinctes pour les utilisateurs basiques, intermédiaires et avancés de Studio à l'aide des exemples de politiques présentés sur cette page.

Le tableau Autorisations AWS Identity and Access Management pour les utilisateurs d'EMR Studio détaille chaque opération Studio qu'un utilisateur peut effectuer et répertorie les actions IAM minimales requises pour les effectuer. Pour savoir comment créer des politiques, consultez la rubrique Création de politiques IAM du Guide de l'utilisateur IAM.

Votre politique d'autorisations doit inclure les déclarations suivantes.

```
{
            "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect": "Allow",
            "Action": "secretsmanager:TagResource",
            "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
},
{
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/your-emr-studio-service-role"
            ],
            "Effect": "Allow"
}
```

Définir la propriété pour la collaboration dans l'espace de travail

La collaboration dans l'espace de travail permet à plusieurs utilisateurs de travailler simultanément dans le même espace de travail et peut être configurée à l'aide du panneau Collaboration de l'interface utilisateur de l'espace de travail. Pour voir et utiliser le panneau Collaboration, un utilisateur doit disposer des autorisations suivantes. Tout utilisateur disposant de ces autorisations peut voir et utiliser le panneau Collaboration.

```
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce:DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities"
```

Pour limiter l'accès au panneau Collaboration, vous pouvez utiliser le contrôle d'accès basé sur les balises. Lorsqu'un utilisateur crée un espace de travail, EMR Studio applique une balise par défaut avec une clé `creatorUserId` dont la valeur est l'ID de l'utilisateur qui crée l'espace de travail.

> **Note**
>
> EMR Studio ajoute la balise `creatorUserId` aux Workspaces créés après le 16 novembre 2021. Pour limiter les personnes autorisées à configurer la collaboration pour les Workspaces créés avant cette date, nous vous recommandons d’ajouter manuellement

la balise `creatorUserId` à votre Workspace, puis d'utiliser le contrôle d'accès basé sur les balises dans vos politiques d'autorisations utilisateur.

L'exemple d'instruction suivant permet à un utilisateur de configurer la collaboration pour n'importe quel espace de travail avec la clé de balise `creatorUserId` dont la valeur correspond à l'ID de l'utilisateur (indiqué par la variable de politique `aws:userId`). En d'autres termes, l'instruction permet à un utilisateur de configurer la collaboration pour les espaces de travail qu'il crée. Pour en savoir plus sur les variables de politique, voir la rubrique Éléments des politiques IAM : variables et balises du Guide de l'utilisateur IAM.

```
{
    "Sid": "UserRolePermissionsForCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce:DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
        }
    }
}
```

Création d'une politique de secrets Git de niveau utilisateur

Rubriques

- Pour utiliser les autorisations au niveau de l'utilisateur
- Pour passer des autorisations au niveau du service à des autorisations au niveau de l'utilisateur
- Pour utiliser les autorisations au niveau du service

Pour utiliser les autorisations au niveau de l'utilisateur

EMR Studio ajoute automatiquement la balise `for-use-with-amazon-emr-managed-user-policies` lorsqu'il crée des secrets Git. Pour contrôler l'accès aux secrets Git au niveau utilisateur,

ajoutez des autorisations basées sur les balises à la politique de rôle d'utilisateur EMR Studio avec `secretsmanager:GetSecretValue`, comme indiqué dans section Pour passer des autorisations au niveau du service à des autorisations au niveau de l'utilisateur ci-dessous.

Si la politique de fonction du service EMR Studio comprend des autorisations pour `secretsmanager:GetSecretValue`, supprimez-les.

Pour passer des autorisations au niveau du service à des autorisations au niveau de l'utilisateur

> **Note**
>
> La balise `for-use-with-amazon-emr-managed-user-policies` garantit que les autorisations de l'étape 1 ci-dessous accordent au créateur du Workspace l'accès au secret Git. Toutefois, si vous avez lié des référentiels Git avant le 1er septembre 2023, l'accès aux secrets Git correspondants sera refusé, car aucune balise `for-use-with-amazon-emr-managed-user-policies` n'y est appliquée. Pour appliquer des autorisations au niveau utilisateur, vous devez recréer les anciens secrets à partir des référentiels Git appropriés JupyterLab et les lier à nouveau.
> Pour plus d'informations sur les variables de politique, voir la rubrique Éléments des politiques IAM : variables et balises du Guide de l'utilisateur IAM.

1. Ajoutez les autorisations suivantes à la politique de rôle d'utilisateur EMR Studio. La politique utilise la clé `for-use-with-amazon-emr-managed-user-policies` avec une valeur `"${aws:userid}"`.

```
{
   "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:*:*:secret:*",
    "Condition": {
        "StringEquals": {
            "secretsmanager:ResourceTag/for-use-with-amazon-emr-managed-user-
policies": "${aws:userid}"
        }
    }
}
```

2. Si elle existe, supprimez l'autorisation suivante de la politique de fonction du service EMR Studio. La politique de fonction du service s'appliquant à tous les secrets définis par chaque utilisateur, vous ne devez effectuer cette opération qu'une seule fois.

```
{
    "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
     ],
    "Resource": "arn:aws:secretsmanager:*:*:secret:*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
    }
}
```

Pour utiliser les autorisations au niveau du service

À compter du 1er septembre 2023, EMR Studio ajoute automatiquement la balise `for-use-with-amazon-emr-managed-user-policies` pour le contrôle d'accès au niveau de l'utilisateur. Il s'agit d'une fonctionnalité supplémentaire. Vous pouvez donc continuer à utiliser l'accès au niveau du service disponible via l'autorisation `GetSecretValue` associée à la fonction du service EMR Studio.

EMR Studio n'a pas ajouté la balise `for-use-with-amazon-emr-managed-user-policies` pour les secrets créés avant le 1er septembre 2023. Pour continuer à utiliser les autorisations au niveau du service, il vous suffit de conserver votre fonction du service EMR Studio et vos autorisations de rôle utilisateur. Toutefois, pour limiter les personnes autorisées à accéder à un secret, nous vous recommandons de suivre les étapes de la rubrique Pour utiliser les autorisations au niveau de l'utilisateur pour ajouter manuellement la balise `for-use-with-amazon-emr-managed-user-policies` à vos secrets, puis d'utiliser le contrôle d'accès basé sur les balises dans vos politiques d'autorisations utilisateur.

Pour plus d'informations sur les variables de politique, voir la rubrique Éléments des politiques IAM : variables et balises du Guide de l'utilisateur IAM.

Associez les politiques d’autorisations à votre identité IAM.

Le tableau suivant récapitule l'identité IAM à laquelle vous attachez une politique d'autorisations, en fonction de votre mode d'authentification EMR Studio. Pour obtenir des instructions sur la façon d'attacher une politique, consultez Ajout et suppression d'autorisations basées sur l'identité IAM.

Si vous utilisez...	Attachez la politique à...
Authentification IAM	Vos identités IAM (utilisateurs, groupes d'utilisateurs ou rôles). Par exemple, vous pouvez attacher une politique d'autorisations à un utilisateur de votre Compte AWS.
Fédération IAM avec un fournisseur d'identité (IdP) externe	Le ou les rôles IAM que vous créez pour votre IdP externe. Par exemple, une fédération IAM pour SAML 2.0. EMR Studio utilise les autorisations que vous attachez à vos rôles IAM pour les utilisateurs disposant d'un accès fédéré à un Studio.
IAM Identity Center	Votre rôle d'utilisateur Amazon EMR Studio.

Exemple de politiques utilisateur

La politique utilisateur basique suivante autorise la plupart des actions EMR Studio, mais ne permet pas à un utilisateur de créer de nouveaux clusters Amazon EMR.

Politique basique

> **Important**
>
> L'exemple de politique n'inclut pas l'autorisation `CreateStudioPresignedUrl`, que vous devez accorder à un utilisateur lorsque vous utilisez le mode d'authentification IAM. Pour plus d’informations, consultez Attribuer un utilisateur ou un groupe à un EMR Studio.

L'exemple de politique inclut des éléments `Condition` visant à appliquer le contrôle d'accès basé sur les balises (TBAC) afin que vous puissiez utiliser la politique avec l'exemple de fonction du

service pour EMR Studio. Pour plus d'informations, consultez Créer une fonction du service EMR Studio.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSecurityGroup"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:vpc/*"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/for-use-with-amazon-emr-managed-policies":"true"
            }
         }
      },
      {
         "Sid":"AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
         "Effect":"Allow",
         "Action":[
            "ec2:CreateTags"
         ],
         "Resource":"arn:aws:ec2:*:*:security-group/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true",
               "ec2:CreateAction":"CreateSecurityGroup"
            }
         }
      },
      {
         "Sid":"AllowSecretManagerListSecrets",
         "Action":[
            "secretsmanager:ListSecrets"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
```

```
    {
       "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
       "Effect":"Allow",
       "Action":"secretsmanager:CreateSecret",
       "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
       "Condition":{
          "StringEquals":{
             "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
          }
       }
    },
    {
       "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
       "Effect":"Allow",
       "Action":"secretsmanager:TagResource",
       "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
    },
    {
       "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
       "Action":"iam:PassRole",
       "Resource":[
          "arn:aws:iam::*:role/<your-emr-studio-service-role>"
       ],
       "Effect":"Allow"
    },
    {
       "Sid":"AllowS3ListAndLocationPermissions",
       "Action":[
          "s3:ListAllMyBuckets",
          "s3:ListBucket",
          "s3:GetBucketLocation"
       ],
       "Resource":"arn:aws:s3:::*",
       "Effect":"Allow"
    },
    {
       "Sid":"AllowS3ReadOnlyAccessToLogs",
       "Action":[
          "s3:GetObject"
       ],
       "Resource":[
          "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
       ],
       "Effect":"Allow"
```

```
      },
      {
         "Sid":"AllowConfigurationForWorkspaceCollaboration",
         "Action":[
            "elasticmapreduce:UpdateEditor",
            "elasticmapreduce:PutWorkspaceAccess",
            "elasticmapreduce:DeleteWorkspaceAccess",
            "elasticmapreduce:ListWorkspaceAccessIdentities"
         ],
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringEquals":{
               "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
            }
         }
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      }
   ]
}
```

La politique utilisateur intermédiaire suivante autorise la plupart des actions EMR Studio et permet à un utilisateur de créer de nouveaux clusters Amazon EMR à l'aide d'un modèle de cluster.

Politique intermédiaire

> **Important**
>
> L'exemple de politique n'inclut pas l'autorisation `CreateStudioPresignedUrl`, que vous devez accorder à un utilisateur lorsque vous utilisez le mode d'authentification IAM. Pour plus d'informations, consultez Attribuer un utilisateur ou un groupe à un EMR Studio.

L'exemple de politique inclut des éléments `Condition` visant à appliquer le contrôle d'accès basé sur les balises (TBAC) afin que vous puissiez utiliser la politique avec l'exemple de fonction du service pour EMR Studio. Pour plus d'informations, consultez Créer une fonction du service EMR Studio.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowEMRBasicActions",
         "Action":[
            "elasticmapreduce:CreateEditor",
            "elasticmapreduce:DescribeEditor",
            "elasticmapreduce:ListEditors",
            "elasticmapreduce:StartEditor",
            "elasticmapreduce:StopEditor",
            "elasticmapreduce:DeleteEditor",
            "elasticmapreduce:OpenEditorInConsole",
            "elasticmapreduce:AttachEditor",
            "elasticmapreduce:DetachEditor",
            "elasticmapreduce:CreateRepository",
            "elasticmapreduce:DescribeRepository",
            "elasticmapreduce:DeleteRepository",
            "elasticmapreduce:ListRepositories",
            "elasticmapreduce:LinkRepository",
            "elasticmapreduce:UnlinkRepository",
            "elasticmapreduce:DescribeCluster",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListSteps",
            "elasticmapreduce:CreatePersistentAppUI",
            "elasticmapreduce:DescribePersistentAppUI",
```

```
            "elasticmapreduce:GetPersistentAppUIPresignedURL",
            "elasticmapreduce:GetOnClusterAppUIPresignedURL"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowEMRContainersBasicActions",
         "Action":[
            "emr-containers:DescribeVirtualCluster",
            "emr-containers:ListVirtualClusters",
            "emr-containers:DescribeManagedEndpoint",
            "emr-containers:ListManagedEndpoints",
            "emr-containers:DescribeJobRun",
            "emr-containers:ListJobRuns"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
          "Sid": "AllowRetrievingManagedEndpointCredentials",
          "Effect": "Allow",
          "Action": [
              "emr-containers:GetManagedEndpointSessionCredentials"
          ],
          "Resource": [
              "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
          ],
          "Condition": {
              "StringEquals": {
                  "emr-containers:ExecutionRoleArn": [
                      "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                  ]
              }
          }
      },
      {
         "Sid":"AllowSecretManagerListSecrets",
         "Action":[
            "secretsmanager:ListSecrets"
         ],
         "Resource":"*",
         "Effect":"Allow"
```

```
        },
        {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
                "StringEquals":{
                    "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
                }
            }
        },
        {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid":"AllowClusterTemplateRelatedIntermediateActions",
            "Action":[
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:ProvisionProduct",
                "servicecatalog:SearchProducts",
                "servicecatalog:UpdateProvisionedProduct",
                "servicecatalog:ListProvisioningArtifacts",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:DescribeRecord",
                "cloudformation:DescribeStackResources"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
            "Action":"iam:PassRole",
            "Resource":[
                "arn:aws:iam::*:role/<your-emr-studio-service-role>"
            ],
            "Effect":"Allow"
        },
        {
```

```
        "Sid":"AllowS3ListAndLocationPermissions",
        "Action":[
           "s3:ListAllMyBuckets",
           "s3:ListBucket",
           "s3:GetBucketLocation"
        ],
        "Resource":"arn:aws:s3:::*",
        "Effect":"Allow"
     },
     {
        "Sid":"AllowS3ReadOnlyAccessToLogs",
        "Action":[
           "s3:GetObject"
        ],
        "Resource":[
           "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
        ],
        "Effect":"Allow"
     },
     {
        "Sid":"AllowConfigurationForWorkspaceCollaboration",
        "Action":[
           "elasticmapreduce:UpdateEditor",
           "elasticmapreduce:PutWorkspaceAccess",
           "elasticmapreduce:DeleteWorkspaceAccess",
           "elasticmapreduce:ListWorkspaceAccessIdentities"
        ],
        "Resource":"*",
        "Effect":"Allow",
        "Condition":{
           "StringEquals":{
              "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
           }
        }
     },
     {
        "Sid":"DescribeNetwork",
        "Effect":"Allow",
        "Action":[
           "ec2:DescribeVpcs",
           "ec2:DescribeSubnets",
           "ec2:DescribeSecurityGroups"
        ],
        "Resource":"*"
```

```
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      },
      {
         "Sid": "AllowServerlessActions",
         "Action": [
            "emr-serverless:CreateApplication",
            "emr-serverless:UpdateApplication",
            "emr-serverless:DeleteApplication",
            "emr-serverless:ListApplications",
            "emr-serverless:GetApplication",
            "emr-serverless:StartApplication",
            "emr-serverless:StopApplication",
            "emr-serverless:StartJobRun",
            "emr-serverless:CancelJobRun",
            "emr-serverless:ListJobRuns",
            "emr-serverless:GetJobRun",
            "emr-serverless:GetDashboardForJobRun",
            "emr-serverless:AccessInteractiveEndpoints"
         ],
         "Resource": "*",
         "Effect": "Allow"
        },
        {
           "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
           "Action": "iam:PassRole",
           "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
           "Effect": "Allow"
        }
   ]
}
```

La politique utilisateur avancée suivante autorise toutes les actions EMR Studio et permet à un utilisateur de créer de nouveaux clusters Amazon EMR à l'aide d'un modèle de cluster ou en fournissant une configuration de cluster.

Politique avancée

> **Important**
>
> L'exemple de politique n'inclut pas l'autorisation `CreateStudioPresignedUrl`, que vous devez accorder à un utilisateur lorsque vous utilisez le mode d'authentification IAM. Pour plus d'informations, consultez Attribuer un utilisateur ou un groupe à un EMR Studio.

L'exemple de politique inclut des éléments `Condition` visant à appliquer le contrôle d'accès basé sur les balises (TBAC) afin que vous puissiez utiliser la politique avec l'exemple de fonction du service pour EMR Studio. Pour plus d'informations, consultez Créer une fonction du service EMR Studio.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowEMRBasicActions",
         "Action":[
            "elasticmapreduce:CreateEditor",
            "elasticmapreduce:DescribeEditor",
            "elasticmapreduce:ListEditors",
            "elasticmapreduce:StartEditor",
            "elasticmapreduce:StopEditor",
            "elasticmapreduce:DeleteEditor",
            "elasticmapreduce:OpenEditorInConsole",
            "elasticmapreduce:AttachEditor",
            "elasticmapreduce:DetachEditor",
            "elasticmapreduce:CreateRepository",
            "elasticmapreduce:DescribeRepository",
            "elasticmapreduce:DeleteRepository",
            "elasticmapreduce:ListRepositories",
            "elasticmapreduce:LinkRepository",
            "elasticmapreduce:UnlinkRepository",
            "elasticmapreduce:DescribeCluster",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListSteps",
            "elasticmapreduce:CreatePersistentAppUI",
            "elasticmapreduce:DescribePersistentAppUI",
```

```
            "elasticmapreduce:GetPersistentAppUIPresignedURL",
            "elasticmapreduce:GetOnClusterAppUIPresignedURL"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowEMRContainersBasicActions",
         "Action":[
            "emr-containers:DescribeVirtualCluster",
            "emr-containers:ListVirtualClusters",
            "emr-containers:DescribeManagedEndpoint",
            "emr-containers:ListManagedEndpoints",
            "emr-containers:DescribeJobRun",
            "emr-containers:ListJobRuns"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
          "Sid": "AllowRetrievingManagedEndpointCredentials",
          "Effect": "Allow",
          "Action": [
              "emr-containers:GetManagedEndpointSessionCredentials"
          ],
          "Resource": [
              "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
          ],
          "Condition": {
              "StringEquals": {
                  "emr-containers:ExecutionRoleArn": [
                      "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                  ]
              }
          }
      },
      {
         "Sid":"AllowSecretManagerListSecrets",
         "Action":[
            "secretsmanager:ListSecrets"
         ],
         "Resource":"*",
         "Effect":"Allow"
```

```
        },
        {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
               }
            }
        },
        {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid":"AllowClusterTemplateRelatedIntermediateActions",
            "Action":[
               "servicecatalog:DescribeProduct",
               "servicecatalog:DescribeProductView",
               "servicecatalog:DescribeProvisioningParameters",
               "servicecatalog:ProvisionProduct",
               "servicecatalog:SearchProducts",
               "servicecatalog:UpdateProvisionedProduct",
               "servicecatalog:ListProvisioningArtifacts",
               "servicecatalog:ListLaunchPaths",
               "servicecatalog:DescribeRecord",
               "cloudformation:DescribeStackResources"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowEMRCreateClusterAdvancedActions",
            "Action":[
               "elasticmapreduce:RunJobFlow"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
        {
```

```
            "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
            "Action":"iam:PassRole",
            "Resource":[
               "arn:aws:iam::*:role/<your-emr-studio-service-role>",
               "arn:aws:iam::*:role/EMR_DefaultRole_V2",
               "arn:aws:iam::*:role/EMR_EC2_DefaultRole"
            ],
            "Effect":"Allow"
         },
         {
            "Sid":"AllowS3ListAndLocationPermissions",
            "Action":[
               "s3:ListAllMyBuckets",
               "s3:ListBucket",
               "s3:GetBucketLocation"
            ],
            "Resource":"arn:aws:s3:::*",
            "Effect":"Allow"
         },
         {
            "Sid":"AllowS3ReadOnlyAccessToLogs",
            "Action":[
               "s3:GetObject"
            ],
            "Resource":[
               "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
            ],
            "Effect":"Allow"
         },
         {
            "Sid":"AllowConfigurationForWorkspaceCollaboration",
            "Action":[
               "elasticmapreduce:UpdateEditor",
               "elasticmapreduce:PutWorkspaceAccess",
               "elasticmapreduce:DeleteWorkspaceAccess",
               "elasticmapreduce:ListWorkspaceAccessIdentities"
            ],
            "Resource":"*",
            "Effect":"Allow",
            "Condition":{
               "StringEquals":{
                  "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
               }
            }
```

```
    },
    {
       "Sid" : "SageMakerDataWranglerForEMRStudio",
       "Effect" : "Allow",
       "Action" : [
          "sagemaker:CreatePresignedDomainUrl",
          "sagemaker:DescribeDomain",
          "sagemaker:ListDomains",
          "sagemaker:ListUserProfiles"
       ],
       "Resource":"*"
    },
    {
       "Sid":"DescribeNetwork",
       "Effect":"Allow",
       "Action":[
          "ec2:DescribeVpcs",
          "ec2:DescribeSubnets",
          "ec2:DescribeSecurityGroups"
       ],
       "Resource":"*"
    },
    {
       "Sid":"ListIAMRoles",
       "Effect":"Allow",
       "Action":[
          "iam:ListRoles"
       ],
       "Resource":"*"
    },
    {
       "Sid": "AllowServerlessActions",
       "Action": [
          "emr-serverless:CreateApplication",
          "emr-serverless:UpdateApplication",
          "emr-serverless:DeleteApplication",
          "emr-serverless:ListApplications",
          "emr-serverless:GetApplication",
          "emr-serverless:StartApplication",
          "emr-serverless:StopApplication",
          "emr-serverless:StartJobRun",
          "emr-serverless:CancelJobRun",
          "emr-serverless:ListJobRuns",
          "emr-serverless:GetJobRun",
```

```
            "emr-serverless:GetDashboardForJobRun",
            "emr-serverless:AccessInteractiveEndpoints"
         ],
         "Resource": "*",
         "Effect": "Allow"
      },
      {
         "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
         "Action": "iam:PassRole",
         "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
         "Effect": "Allow"
      },
      {
         "Sid": "AllowCodeWhisperer",
         "Effect": "Allow",
         "Action": [ "codewhisperer:GenerateRecommendations" ],
         "Resource": "*"
      },
      {
         "Sid": "AllowAthenaSQL",
         "Action": [
            "athena:StartQueryExecution",
            "athena:StopQueryExecution",
            "athena:GetQueryExecution",
            "athena:GetQueryRuntimeStatistics",
            "athena:GetQueryResults",
            "athena:ListQueryExecutions",
            "athena:BatchGetQueryExecution",
            "athena:GetNamedQuery",
            "athena:ListNamedQueries",
            "athena:BatchGetNamedQuery",
            "athena:UpdateNamedQuery",
            "athena:DeleteNamedQuery",
            "athena:ListDataCatalogs",
            "athena:GetDataCatalog",
            "athena:ListDatabases",
            "athena:GetDatabase",
            "athena:ListTableMetadata",
            "athena:GetTableMetadata",
            "athena:ListWorkGroups",
            "athena:GetWorkGroup",
            "athena:CreateNamedQuery",
            "athena:GetPreparedStatement",
            "glue:CreateDatabase",
```

```
            "glue:DeleteDatabase",
            "glue:GetDatabase",
            "glue:GetDatabases",
            "glue:UpdateDatabase",
            "glue:CreateTable",
            "glue:DeleteTable",
            "glue:BatchDeleteTable",
            "glue:UpdateTable",
            "glue:GetTable",
            "glue:GetTables",
            "glue:BatchCreatePartition",
            "glue:CreatePartition",
            "glue:DeletePartition",
            "glue:BatchDeletePartition",
            "glue:UpdatePartition",
            "glue:GetPartition",
            "glue:GetPartitions",
            "glue:BatchGetPartition",
            "kms:ListAliases",
            "kms:ListKeys",
            "kms:DescribeKey",
            "lakeformation:GetDataAccess",
            "s3:GetBucketLocation",
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts",
            "s3:AbortMultipartUpload",
            "s3:PutObject",
            "s3:PutBucketPublicAccessBlock",
            "s3:ListAllMyBuckets"
        ],
        "Resource": "*",
        "Effect": "Allow"
      }
   ]
}
```

La politique utilisateur suivante définit les autorisations utilisateur minimales requises pour utiliser une application interactive EMR sans serveur avec les espaces de travail EMR Studio.

Politique interactive EMR sans serveur

Dans cet exemple de politique qui prévoit des autorisations utilisateur pour les applications interactives EMR Serverless avec EMR Studio, remplacez les espaces réservés pour et par votre rôle de service EMR Studio et *emr-studio-service-role*votre rôle *serverless-runtime-role*d'exécution EMR Serverless corrects.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServerlessActions",
            "Action": [
                "emr-serverless:CreateApplication",
                "emr-serverless:UpdateApplication",
                "emr-serverless:DeleteApplication",
                "emr-serverless:ListApplications",
                "emr-serverless:GetApplication",
                "emr-serverless:StartApplication",
                "emr-serverless:StopApplication",
                "emr-serverless:StartJobRun",
                "emr-serverless:CancelJobRun",
                "emr-serverless:ListJobRuns",
                "emr-serverless:GetJobRun",
                "emr-serverless:GetDashboardForJobRun",
                "emr-serverless:AccessInteractiveEndpoints"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowEMRBasicActions",
            "Action": [
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:UpdateStudio",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:AttachEditor",
                "elasticmapreduce:DetachEditor",
```

```
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingRuntimeRoleForRunningEMRServerlessJob",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/emr-studio-service-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowS3ListAndGetPermissions",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
            "Sid":"DescribeNetwork",
            "Effect":"Allow",
            "Action":[
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource":"*"
        },
        {
            "Sid": "ListIAMRoles",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

Autorisations AWS Identity and Access Management pour les utilisateurs d'EMR Studio

Le tableau suivant inclut chaque opération Amazon EMR Studio qu'un utilisateur peut effectuer et répertorie les actions IAM minimales requises pour effectuer cette opération. Vous autorisez ces actions dans vos politiques d'autorisations IAM (lorsque vous utilisez l'authentification IAM) ou dans vos politiques de session de rôle d'utilisateur (lorsque vous utilisez l'authentification IAM Identity Center) pour EMR Studio.

Le tableau affiche également les opérations autorisées dans chacun des exemples de politique d'autorisations pour EMR Studio. Pour plus d'informations sur les exemples de politiques d'autorisations, consultez Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio.

Action	Base	Intermédiaire	Avancé	Actions associées
Créer et supprimer des espaces de travail	Oui	Oui	Oui	`"elasticmapreduce:CreateEditor", "elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:DeleteEditor"`
Afficher le panneau Collaboration, activer la collaboration dans l'espace de travail et ajouter des collaborateurs. Pour plus d'informations, consultez Définir la propriété pour	Oui	Oui	Oui	`"elasticmapreduce:UpdateEditor", "elasticmapreduce:PutWorkspaceAccess", "elasticmapreduce:DeleteWorkspaceAccess",`

Action	Base	Intermédiaire	Avancé	Actions associées
la collaboration dans le Workspace.				`"elasticmapreduce:ListWorkspaceAccessIdentities"`
Consulter une liste des compartiments de stockage Amazon S3 Control dans le même compte que le Studio lors de la création d'un cluster EMR, et accéder aux journaux des conteneurs lors de l'utilisation d'une interface utilisateur web pour déboguer des applications	Oui	Oui	Oui	`"s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetBucketLocation", "s3:GetObject"`
Accéder aux espaces de travail	Oui	Oui	Oui	`"elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:StartEditor", "elasticmapreduce:StopEditor", "elasticmapreduce:OpenEditorInConsole"`

Action	Base	Intermédiaire	Avancé	Actions associées
Attacher ou détacher des clusters Amazon EMR existants associés à l'espace de travail	Oui	Oui	Oui	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListInstanceGroups", "elasticmapreduce:ListBootstrapActions"`
Attacher ou détacher Amazon EMR sur des clusters EKS	Oui	Oui	Oui	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListManagedEndpoints", "emr-containers:DescribeManagedEndpoint", "emr-containers:GetManagedEndpointSessionCredentials"`

Action	Base	Intermédiaire	Avancé	Actions associées
Attacher ou détacher les applications EMR sans serveur associées à l'espace de travail	Non	Oui	Oui	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-serverless:GetApplication", "emr-serverless:StartApplication", "emr-serverless:ListApplications", "emr-serverless:GetDashboardForJobRun", "emr-serverless:AccessInteractiveEndpoints", "iam:PassRole"` L'autorisation `PassRole` est requise pour transmettre le rôle d'exécution de tâches EMR sans serveur. Pour plus d'informations, veuillez consulter la rubrique Job runtime roles dans le Guide de l'utilisateur Amazon EMR sans serveur.

Action	Base	Intermédiaire	Avancé	Actions associées
Déboguer Amazon EMR sur des tâches EC2 avec des interfaces utilisateur d'application persistantes	Oui	Oui	Oui	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "elasticmapreduce:ListClusters", "elasticmapreduce:ListSteps", "elasticmapreduce:DescribeCluster", "s3:ListBucket", "s3:GetObject"`
Déboguer Amazon EMR sur des tâches EC2 avec des interfaces utilisateur d'application intégrées au cluster	Oui	Oui	Oui	`"elasticmapreduce:GetOnClusterAppUIPresignedURL"`

Action	Base	Intermédiaire	Avancé	Actions associées
Déboguer Amazon EMR sur les tâches EKS exécutées à l'aide du serveur d'historique Spark	Oui	Oui	Oui	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListJobRuns", "emr-containers:DescribeJobRun", "s3:ListBucket", "s3:GetObject"`
Créer et supprimer des référentiels Git	Oui	Oui	Oui	`"elasticmapreduce:CreateRepository", "elasticmapreduce:DeleteRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository", "secretsmanager:CreateSecret", "secretsmanager:ListSecrets", "secretsmanager:TagResource"`

Action	Base	Intermédiaire	Avancé	Actions associées
Créer ou annuler des liens à des référentiels Git	Oui	Oui	Oui	`"elasticmapreduce:LinkRepository", "elasticmapreduce:UnlinkRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository"`
Créer des clusters à partir de modèles de clusters prédéfinis	Non	Oui	Oui	`"servicecatalog:SearchProducts", "servicecatalog:DescribeProduct", "servicecatalog:DescribeProductView", "servicecatalog:DescribeProvisioningParameters", "servicecatalog:ProvisionProduct", "servicecatalog:UpdateProvisionedProduct", "servicecatalog:ListProvisioningArtifacts", "servicecatalog:DescribeRecord", "servicecatalog:ListLaunchPaths", "cloudformation:DescribeStackResources", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`

Action	Base	Intermédiaire	Avancé	Actions associées
Spécifier une configuration de cluster pour créer de nouveaux clusters	Non	Non	Oui	`"elasticmapreduce:RunJobFlow", "iam:PassRole", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`
Affecter un utilisateur à un Studio lorsque vous utilisez le mode d'authentification IAM	Non	Non	Non	`"elasticmapreduce:CreateStudioPresignedUrl"`
Décrire les objets réseau.	Oui	Oui	Oui	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "DescribeNetwork", "Effect": "Allow", "Action": [ "ec2:DescribeVpcs", "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups" ], "Resource": "*" } ] }`

Action	Base	Intermédiaire	Avancé	Actions associées
Répertorier les rôles IAM.	Oui	Oui	Oui	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "ListIAMRoles", "Effect": "Allow", "Action": [ "iam:ListRoles" ], "Resource": "*" } ] }`
Connectez-vous à EMR Studio depuis Amazon SageMaker Studio et utilisez l'interface visuelle Data Wrangler.	Non	Non	Oui	`"sagemaker:CreatePresignedDomainUrl", "sagemaker:DescribeDomain", "sagemaker:ListDomains", "sagemaker:ListUserProfiles"`
Utilisez Amazon CodeWhisperer dans votre studio EMR.	Non	Non	Oui	`"codewhisperer:GenerateRecommendations"`

Action	Base	Intermédiaire	Avancé	Actions associées
Accéder à l'éditeur SQL Amazon Athena depuis votre Studio EMR Cette liste peut ne pas inclure toutes les autorisations dont vous avez besoin pour utiliser toutes les fonctionnalités Athena. Pour obtenir la up-to-date liste la plus complète, consultez la politique d'accès complet d'Athena.	Non	Non	Oui	`"athena:StartQueryExecution", "athena:StopQueryExecution", "athena:GetQueryExecution", "athena:GetQueryRuntimeStatistics", "athena:GetQueryResults", "athena:ListQueryExecutions", "athena:BatchGetQueryExecution", "athena:GetNamedQuery", "athena:ListNamedQueries", "athena:BatchGetNamedQuery", "athena:UpdateNamedQuery", "athena:DeleteNamedQuery", "athena:ListDataCatalogs", "athena:GetDataCatalog", "athena:ListDatabases", "athena:GetDatabase", "athena:ListTableMetadata", "athena:GetTableMetadata", "athena:ListWorkGroups", "athena:GetWorkGroup", "athena:CreateNamedQuery", "athena:GetPreparedStatement", "glue:CreateDatabase", "glue:DeleteDatabase", "glue:GetDatabase",`

Action	Base	Intermédiaire	Avancé	Actions associées
				"glue:GetDatabases", "glue:UpdateDatabase", "glue:CreateTable", "glue:DeleteTable", "glue:BatchDeleteTable", "glue:UpdateTable", "glue:GetTable", "glue:GetTables", "glue:BatchCreatePartition", "glue:CreatePartition", "glue:DeletePartition", "glue:BatchDeletePartition", "glue:UpdatePartition", "glue:GetPartition", "glue:GetPartitions", "glue:BatchGetPartition", "kms:ListAliases", "kms:ListKeys", "kms:DescribeKey", "lakeformation:GetDataAccess", "s3:GetBucketLocation", "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload", "s3:PutObject", "s3:PutBucketPublicAccessBlock", "s3:ListAllMyBuckets"

Créer un EMR Studio

Vous pouvez créer un Studio EMR pour votre équipe à l'aide de la console Amazon EMR ou de la AWS CLI. La création d'une instance Studio fait partie de la configuration d'Amazon EMR Studio.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

Prérequis

Avant de créer un Studio, assurez-vous d'avoir effectué les tâches précédentes dans Configurer un Amazon EMR Studio.

Pour créer un Studio à l'aide de l'AWS CLI, vous devez avoir installé la dernière version. Pour plus d'informations, consultez Installation ou mise à jour de la version la plus récente de l'AWS CLI.

> **Important**
>
> Désactivez les outils de gestion de proxy tels que FoxyProxy ou SwitchyOmega dans le navigateur avant de créer un Studio. Les proxys actifs peuvent générer un message d'erreur de défaillance du réseau lorsque vous choisissez Créer un studio.

Amazon EMR vous fournit une expérience de console simple pour créer un studio, afin que vous puissiez rapidement démarrer avec les paramètres par défaut, pour exécuter des charges de travail interactives ou des tâches par lots avec les paramètres par défaut. La création d'un studio EMR crée également une application EMR Serverless prête à exécuter vos tâches interactives.

Si vous souhaitez contrôler totalement les paramètres de votre studio, vous pouvez choisir Personnalisé, qui vous permet de configurer tous les paramètres supplémentaires.

Interactive workloads

Pour créer un studio EMR pour les charges de travail interactives

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

2. Sous EMR Studio dans le menu de navigation de gauche, choisissez Mise en route. Vous pouvez également créer un Studio à partir de la page Studios.
3. Amazon EMR fournit des paramètres par défaut si vous créez un studio EMR pour les charges de travail interactives, mais vous pouvez modifier ces paramètres. Les paramètres configurables incluent le nom du studio EMR, l'emplacement S3 de votre espace de travail, le rôle de service à utiliser, le ou les espaces de travail que vous souhaitez utiliser, le nom de l'application EMR Serverless et le rôle d'exécution associé.
4. Choisissez Create Studio et lancez Workspace pour terminer et accéder à la page Studios. Votre nouveau Studio apparaît dans la liste avec des informations telles que le Nom du Studio, la Date de création et l'URL d'accès Studio. Votre espace de travail s'ouvre dans un nouvel onglet de votre navigateur.

Batch jobs

Pour créer un studio EMR pour les charges de travail interactives

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR Studio dans le menu de navigation de gauche, choisissez Mise en route. Vous pouvez également créer un Studio à partir de la page Studios.
3. Amazon EMR fournit des paramètres par défaut si vous créez un studio EMR pour les tâches par lots, mais vous pouvez modifier ces paramètres. Les paramètres configurables incluent le nom du studio EMR, le nom de l'application EMR Serverless et le rôle d'exécution associé.
4. Choisissez Create Studio et lancez Workspace pour terminer et accéder à la page Studios. Votre nouveau Studio apparaît dans la liste avec des informations telles que le Nom du Studio, la Date de création et l'URL d'accès Studio. Votre EMR Studio s'ouvre dans un nouvel onglet de votre navigateur.

Custom settings

Pour créer un studio EMR avec des paramètres personnalisés

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR Studio dans le menu de navigation de gauche, choisissez Mise en route. Vous pouvez également créer un Studio à partir de la page Studios.
3. Choisissez Créer un Studio pour ouvrir la page Créer un Studio.
4. Entrez le nom du studio.

5. Choisissez de créer un nouveau compartiment S3 ou d'utiliser un emplacement existant.
6. Choisissez l'espace de travail à ajouter au studio. Vous pouvez ajouter jusqu'à 3 espaces de travail.
7. Sous Authentification, choisissez un mode d'authentification pour le Studio et fournissez les informations conformément au tableau suivant. Pour en savoir plus sur l'authentification pour EMR Studio, consultez Choisir un mode d'authentification pour Amazon EMR Studio.

Si vous utilisez...	Faites ceci...
Authentification ou fédération IAM	La méthode d’authentification par défaut est AWS Identity and Access Managemen t (IAM). En bas de l’écran, vous pouvez également ajouter des balises pour permettre à des utilisateurs spécifiques d’accéder au Studio, comme décrit dans la rubrique Attribuer un utilisateur ou un groupe à un EMR Studio. Si vous souhaitez que les utilisateurs fédérés se connectent à l'aide de l'URL Studio et des informations d'identification de votre fournisseur d'identité (IdP), sélectionnez votre IdP dans la liste déroulante, puis entrez l'URL de connexion et le nom du paramètre de votre fournisseur d'identité (IdP). RelayState Pour obtenir la liste des URL et des RelayState noms d'authentification IdP, consultez. RelayState Paramètres du fournisseur d'identité et URL d'authent ification
Authentification IAM Identity Center	Sélectionnez votre Fonction du service et Rôle utilisateur EMR Studio. Pour plus d’informations, consultez Créer une fonction du service EMR Studio et Créer un

Si vous utilisez...	Faites ceci...
	rôle d'utilisateur EMR Studio pour le mode d'authentification IAM Identity Center. Lorsque vous utilisez l'authentification IAM Identity Center (anciennement AWS Single Sign On) pour le Studio, vous pouvez choisir de rationaliser l'expérience de connexion des utilisateurs grâce à l'option Activer la propagation d'identité approuvée. Grâce à la propagation d'identité approuvée , les utilisateurs peuvent se connecter à l'aide de leurs informations d'identification Identity Center et communiquer leur identité aux services AWS en aval lorsqu'ils utilisent le Studio. Dans la section Accès à l'application, vous pouvez également spécifier si tous les utilisateurs et groupes de votre Identity Center doivent avoir accès au Studio, ou si seuls les utilisateurs et groupes assignés que vous choisissez peuvent y accéder. Pour plus d'informations, voir les rubriques Intégrez Amazon EMR à AWS IAM Identity Center et Propagation d'identité approuvée entre applications du Guide de l'utilisateur AWS IAM Identity Center.

8. Pour le VPC, choisissez un Amazon Virtual Private Cloud (VPC) pour le studio dans la liste déroulante.
9. Sous Sous-réseaux, sélectionnez un maximum de cinq sous-réseaux dans votre VPC à associer au Studio. Vous avez la possibilité d'ajouter d'autres sous-réseaux après avoir créé le Studio.

10. Pour Groupes de sécurité, choisissez les groupes de sécurité par défaut ou les groupes de sécurité personnalisés. Pour plus d'informations, consultez Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio.

Si vous choisissez…	Faites ceci...
Les groupes de sécurité EMR Studio par défaut	Pour activer la liaison entre référentiels basée sur Git pour le Studio, choisissez Activer les clusters/points de terminaison et le référentiel Git. Sinon, choisissez Activer les clusters/points de terminaison.
Groupes de sécurité personnalisés pour votre Studio	• Sous Groupe de sécurité du cluster/point de terminaison, sélectionnez le groupe de sécurité moteur que vous avez configuré dans la liste déroulante. Votre Studio utilise ce groupe de sécurité pour autoriser l'accès entrant depuis les espaces de travail attachés. • Sous Groupe de sécurité du cluster/point de terminaison, sélectionnez le groupe de sécurité d'espace de travail que vous avez configuré dans la liste déroulante. Votre Studio utilise ce groupe de sécurité avec les espaces de travail pour fournir un accès sortant aux clusters Amazon EMR attachés et aux référentiels Git hébergés publiquement.

11. Ajoutez des tags à votre Studio et à d'autres ressources. Pour plus d'informations sur les balises, consultez la section Groupes de balises.
12. Choisissez Create Studio et lancez Workspace pour terminer et accéder à la page Studios. Votre nouveau Studio apparaît dans la liste avec des informations telles que le Nom du Studio, la Date de création et l'URL d'accès Studio.

Une fois que vous avez créé un Studio, suivez les instructions de la rubrique Attribuer un utilisateur ou un groupe à un EMR Studio.

CLI

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

Example - Créer un studio EMR qui utilise IAM pour l'authentification

L'exemple de commande AWS CLI suivant crée un EMR Studio avec le mode d'authentification IAM. Lorsque vous utilisez l'authentification ou la fédération IAM pour le Studio, vous ne spécifiez pas de `--user-role`.

Pour permettre aux utilisateurs fédérés de se connecter à l'aide de l'URL Studio et des informations d'identification de votre fournisseur d'identité (IdP), spécifiez votre `--idp-auth-url` et votre `--idp-relay-state-parameter-name`. Pour obtenir la liste des URL et des RelayState noms d'authentification IdP, consultez. RelayState Paramètres du fournisseur d'identité et URL d'authentification

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode IAM \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
--service-role <example-studio-service-role-name> \
--user-role studio-user-role-name \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location> \
--idp-auth-url <https://EXAMPLE/login/> \
--idp-relay-state-parameter-name <example-RelayState>
```

Example - Créer un Studio EMR qui utilise Identity Center pour l'authentification

L'exemple de commande AWS CLI suivant crée un EMR Studio avec le mode d'authentification IAM Identity Center. Lorsque vous utilisez l'authentification IAM Identity Center, vous devez spécifier un `--user-role`.

Pour plus d'informations sur l'authentification IAM Identity Center, consultez Configurer le mode d'authentification IAM Identity Center pour Amazon EMR Studio.

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode SSO \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
--service-role <example-studio-service-role-name> \
--user-role <example-studio-user-role-name> \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location>
--trusted-identity-propagation-enabled \
--idc-user-assignment OPTIONAL \
--idc-instance-arn <iam-identity-center-instance-arn>
```

Example - Sortie CLI pour **`aws emr create-studio`**

Voici un exemple de la sortie qui apparaît après avoir créé un Studio.

```
{
    StudioId: "es-123XXXXXXXXX",
    Url: "https://es-123XXXXXXXXX.emrstudio-prod.us-east-1.amazonaws.com"
}
```

Pour plus d'informations sur la commande `create-studio`, consultez la Référence de commande de l'AWS CLI.

RelayState Paramètres du fournisseur d'identité et URL d'authentification

Lorsque vous utilisez la fédération IAM et que vous souhaitez que les utilisateurs se connectent à l'aide de l'URL de votre studio et des informations d'identification de votre fournisseur d'identité (IdP), vous pouvez spécifier l'URL de connexion RelayStateet le nom du paramètre de votre fournisseur d'identité (IdP) lorsque vous le souhaitez. Créer un EMR Studio

Le tableau suivant indique l'URL d'authentification standard et le nom du RelayState paramètre pour certains fournisseurs d'identité populaires.

Fournisseur d'identité	Paramètre	URL d'authentification
Auth0	`RelayState`	`https://<sub_domain> .auth0.com/samlp/<app_id>`
Comptes Google	`RelayState`	`https://accounts.google.com/o/saml2/initsso?idpid= <idp_id>&spid=<sp_id>&forceauthn=false`
Microsoft Azure	`RelayState`	`https://myapps.microsoft.com/signin/ <app_name> /<app_id>?tenantId= <tenant_id>`
Okta	`RelayState`	`https://<sub_domain> .okta.com/app/<app_name> /<app_id>/sso/saml`
PingFederate	`TargetResource`	`https://<host>/idp/<idp_id>/startSSO.ping?PartnerSpId= <sp_id>`
PingOne	`TargetResource`	`https://sso.connect.pingidentity.com/sso/sp/initsso?saasid= <app_id>&idpid=<idp_id>`

Attribuer et gérer les utilisateurs d'EMR Studio

Après avoir créé un EMR Studio, vous pouvez lui attribuer des utilisateurs et des groupes. La méthode que vous utilisez pour attribuer, mettre à jour et supprimer des utilisateurs dépend du mode d'authentification Studio.

- Lorsque vous utilisez le mode d'authentification IAM, vous configurez l'attribution des utilisateurs et les autorisations EMR Studio dans IAM ou avec IAM et votre fournisseur d'identité.
- Avec le mode d'authentification IAM Identity Center, vous utilisez la console de gestion Amazon EMR ou l'AWS CLI pour gérer les utilisateurs.

Pour en savoir plus sur l'authentification pour Amazon EMR Studio, consultez Choisir un mode d'authentification pour Amazon EMR Studio.

Attribuer un utilisateur ou un groupe à un EMR Studio

IAM

Lorsque vous utilisez Configurer le mode d'authentification IAM pour Amazon EMR Studio, vous devez autoriser l'action `CreateStudioPresignedUrl` dans la politique d'autorisations IAM d'un utilisateur et limiter l'utilisateur à un Studio spécifique. Vous pouvez inclure `CreateStudioPresignedUrl` dans votre Autorisations utilisateur pour le mode d'authentification IAM ou utiliser une politique distincte.

Pour limiter un utilisateur à un Studio (ou à un ensemble de Studios), vous pouvez utiliser le Contrôle d'accès par attributs (ABAC) ou spécifier l'Amazon Resource Name (ARN) d'un Studio dans l'élément `Resource` de la politique d'autorisations.

Example Attribuer un utilisateur à un Studio à l'aide d'un ARN Studio

L'exemple de politique suivant permet à un utilisateur d'accéder à un EMR Studio spécifique en autorisant l'action `CreateStudioPresignedUrl` et en spécifiant l'Amazon Resource Name (ARN) du Studio dans l'élément `Resource`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-
id>:studio/<studio-id>"
        }
    ]
}
```

Example Attribuer un utilisateur à un Studio avec l'ABAC pour l'authentification IAM

Il existe plusieurs méthodes pour configurer le Contrôle d'accès par attributs (ABAC) pour un Studio. Par exemple, vous pouvez attacher une ou plusieurs balises à un EMR Studio, puis

créer une politique IAM qui limite l'action `CreateStudioPresignedUrl` à un Studio ou à un ensemble de Studios spécifiques dotés de ces balises.

Vous pouvez ajouter des balises pendant ou après la création de Studio. Pour ajouter des balises à un Studio existant, utilisez la commande `emr add-tags` de l'AWS CLI. L'exemple suivant ajoute une balise avec la paire clé-valeur `Team = Data Analytics` à un EMR Studio.

```
aws emr add-tags --resource-id <example-studio-id> --tags Team="Data Analytics"
```

L'exemple de politique d'autorisation suivant autorise l'action `CreateStudioPresignedUrl` pour les EMR Studios avec la paire clé-valeur de balise `Team = DataAnalytics`. Pour plus d'informations sur l'utilisation des balises pour le contrôle d'accès, consultez Contrôle de l'accès aux et pour les utilisateurs et rôles IAM à l'aide de balises ou Contrôle de l'accès aux ressources AWS à l'aide de balises.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/Team": "Data Analytics"
                }
            }
        }
    ]
}
```

Example Affecter un utilisateur à un studio à l'aide de la clé de condition SourceIdentity globale aws :

Lorsque vous utilisez la fédération IAM, vous pouvez utiliser la clé de condition globale `aws:SourceIdentity` dans une politique d'autorisations pour donner aux utilisateurs l'accès à Studio lorsqu'ils assument votre rôle IAM pour la fédération.

Vous devez d'abord configurer votre fournisseur d'identité (IdP) pour qu'il renvoie une chaîne d'identification, telle qu'une adresse e-mail ou un nom d'utilisateur, lorsqu'un utilisateur s'authentifie et assume votre rôle IAM pour la fédération. IAM définit la clé de condition globale `aws:SourceIdentity` sur la chaîne d'identification renvoyée par votre IdP.

Pour plus d'informations, consultez le billet de blog Comment associer l'activité des rôles IAM à l'identité d'entreprise dans le blog sur la AWS sécurité et l'SourceIdentityentrée aws : dans la référence des clés de condition globales.

L'exemple de politique suivant autorise l'`CreateStudioPresignedUrl`action et donne aux utilisateurs un accès `aws:SourceIdentity` correspondant au *< example-source-identity >* au studio EMR spécifié par *< example-studio-arn >*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticmapreduce:CreateStudioPresignedUrl",
      "Resource": "<example-studio-arn>",
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": "<example-source-identity>"
        }
      }
    }
  ]
}
```

IAM Identity Center

Lorsque vous attribuez un utilisateur ou un groupe à un EMR Studio, vous devez définir une politique de session qui définit des autorisations détaillées, comme la possibilité de créer un nouveau cluster EMR pour cet utilisateur ou ce groupe. Amazon EMR stocke ces mappages de politiques de session. Vous pouvez mettre à jour la politique de session d'un utilisateur ou d'un groupe après son attribution.

> **Note**
>
> Les autorisations finales pour un utilisateur ou un groupe se situent à l'intersection des autorisations définies dans votre rôle d'utilisateur EMR Studio et des autorisations définies

dans la politique de session pour cet utilisateur ou ce groupe. Si un utilisateur appartient à plusieurs groupes attribués au Studio, EMR Studio utilise une union d'autorisations pour cet utilisateur.

Pour attribuer des utilisateurs ou des groupes à un EMR Studio à l'aide de la console Amazon EMR

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Dans le volet de navigation de gauche, choisissez EMR Studio.
3. Choisissez le nom de votre Studio dans la liste des Studios, ou sélectionnez le Studio et choisissez Afficher les détails pour ouvrir la page des détails du Studio.
4. Choisissez Ajouter des utilisateurs pour voir la table de recherche d'Utilisateurs et de Groupes.
5. Sélectionnez l'onglet Utilisateurs ou Groupes, puis saisissez un terme de recherche dans la barre de recherche pour trouver un utilisateur ou un groupe.
6. Sélectionnez un ou plusieurs utilisateurs ou groupes dans la liste des résultats de recherche. Vous pouvez basculer entre l'onglet Utilisateurs et l'onglet Groupes.
7. Après avoir sélectionné les utilisateurs et les groupes à ajouter au Studio, choisissez Ajouter. Vous devriez voir les utilisateurs et les groupes apparaître dans la liste Utilisateurs Studio. La liste s'actualise au bout de quelques instants.
8. Suivez les instructions dans Mettre à jour les autorisations d'un utilisateur ou d'un groupe attribué à un Studio pour affiner les autorisations Studio pour un utilisateur ou un groupe.

Pour attribuer un utilisateur ou un groupe à un EMR Studio à l'aide de l'AWS CLI

Insérez vos propres valeurs pour les arguments `create-studio-session-mapping` suivants. Pour plus d'informations sur la commande `create-studio-session-mapping`, consultez la Référence de commande de l'AWS CLI.

- **`--studio-id`** : l'ID du Studio auquel vous souhaitez attribuer l'utilisateur ou le groupe. Pour savoir comment récupérer un ID de Studio, consultez la rubrique Afficher les détails de Studio.

- **`--identity-name`** : le nom de l'utilisateur ou du groupe issu de l'Identity Store. Pour plus d'informations, consultez la section UserNamepour les utilisateurs et DisplayNamepour les groupes dans le manuel Identity Store API Reference.
- **`--identity-type`** : utilisez `USER` ou `GROUP` pour spécifier le type d'identité.
- **`--session-policy-arn`** : l'Amazon Resource Name (ARN) de la politique de session que vous souhaitez associer à l'utilisateur ou au groupe. Par exemple, **`arn:aws:iam::`*`<aws-account-id>`*`:policy/`*`EMRStudio_Advanced_User_Policy`**. Pour plus d'informations, consultez Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio.

```
aws emr create-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-name <example-identity-name> \
 --identity-type <USER-or-GROUP> \
 --session-policy-arn <example-session-policy-arn>
```

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

Utilisez la commande `get-studio-session-mapping` pour vérifier la nouvelle attribution. Remplacez *`< example-identity-name >`* par le nom du centre d'identité IAM de l'utilisateur ou du groupe que vous avez mis à jour.

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <user-or-group-name> \
```

Mettre à jour les autorisations d'un utilisateur ou d'un groupe attribué à un Studio

IAM

Pour mettre à jour les autorisations des utilisateurs ou des groupes lorsque vous utilisez le mode d'authentification IAM, utilisez IAM pour modifier les politiques d'autorisations IAM attachées à vos identités IAM (utilisateurs, groupes ou rôles).

Pour plus d'informations, consultez Autorisations utilisateur pour le mode d'authentification IAM.

IAM Identity Center

Pour mettre à jour les autorisations EMR Studio pour un utilisateur ou un groupe à l'aide de la console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Dans le volet de navigation de gauche, choisissez EMR Studio.
3. Choisissez le nom de votre Studio dans la liste des Studios, ou sélectionnez le Studio et choisissez Afficher les détails pour ouvrir la page des détails du Studio.
4. Dans la liste Utilisateurs Studio sur la page détaillée de Studio, recherchez l'utilisateur ou le groupe que vous souhaitez mettre à jour. Vous pouvez effectuer une recherche par nom ou par type d'identité.
5. Sélectionnez l'utilisateur ou le groupe que vous souhaitez mettre à jour et choisissez Attribuer une stratégie pour ouvrir la boîte de dialogue Stratégie de session.
6. Sélectionnez une politique à appliquer à l'utilisateur ou au groupe que vous avez choisi à l'étape 5, puis choisissez Appliquer la stratégie. La liste Utilisateurs Studio doit afficher le nom de la politique dans la colonne Stratégie de session pour l'utilisateur ou le groupe que vous avez mis à jour.

Pour mettre à jour les autorisations EMR Studio pour un utilisateur ou un groupe à l'aide de l'AWS CLI

Insérez vos propres valeurs pour les arguments `update-studio-session-mappings` suivants. Pour plus d'informations sur la commande `update-studio-session-mappings`, consultez la Référence de commande de l'AWS CLI.

```
aws emr update-studio-session-mapping \
```

```
 --studio-id <example-studio-id> \
 --identity-name <name-of-user-or-group-to-update> \
 --session-policy-arn <new-session-policy-arn-to-apply> \
 --identity-type <USER-or-GROUP> \
```

Utilisez la commande `get-studio-session-mapping` pour vérifier l'attribution de la nouvelle politique de session. Remplacez *< example-identity-name >* par le nom du centre d'identité IAM de l'utilisateur ou du groupe que vous avez mis à jour.

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <user-or-group-name> \
```

Supprimer un utilisateur ou un groupe d'un Studio

IAM

Pour supprimer un utilisateur ou un groupe d'un EMR Studio lorsque vous utilisez le mode d'authentification IAM, vous devez révoquer l'accès de l'utilisateur au Studio en reconfigurant la politique d'autorisations IAM de l'utilisateur.

Dans l'exemple de politique suivant, supposons que vous disposiez d'un EMR Studio avec la paire clé-valeur de balise `Team = Quality Assurance`. Selon la politique, l'utilisateur peut accéder aux Studios balisés avec la clé `Team` dont la valeur est égale à `Data Analytics` ou `Quality Assurance`. Pour supprimer l'utilisateur du Studio balisé avec `Team = Quality Assurance`, supprimez `Quality Assurance` de la liste des valeurs de balise.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
```

```
                    "emr:ResourceTag/Team": [
                        "Data Analytics",
                        "Quality Assurance"
                    ]
                }
            }
        }
    ]
}
```

IAM Identity Center

Pour supprimer un utilisateur ou un groupe d'un EMR Studio à l'aide de la console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Dans le volet de navigation de gauche, choisissez EMR Studio.
3. Choisissez le nom de votre Studio dans la liste des Studios, ou sélectionnez le Studio et choisissez Afficher les détails pour ouvrir la page des détails du Studio.
4. Dans la liste Utilisateurs Studio sur la page détaillée de Studio, recherchez l'utilisateur ou le groupe que vous souhaitez supprimer du Studio. Vous pouvez effectuer une recherche par nom ou par type d'identité.
5. Sélectionnez l'utilisateur ou le groupe que vous souhaitez supprimer, puis choisissez Supprimer et confirmez. L'utilisateur ou le groupe que vous avez supprimé disparaît de la liste Utilisateurs Studio.

Pour supprimer un utilisateur ou un groupe d'un EMR Studio à l'aide de l'AWS CLI

Insérez vos propres valeurs pour les arguments `delete-studio-session-mapping` suivants. Pour plus d'informations sur la commande `delete-studio-session-mapping`, consultez la Référence de commande de l'AWS CLI.

```
aws emr delete-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <name-of-user-or-group-to-delete> \
```

Gérer un Amazon EMR Studio

Cette section contient des instructions pour vous aider à surveiller, à mettre à jour ou à supprimer une ressource EMR Studio. Pour plus d'informations sur l'attribution d'utilisateurs ou la mise à jour des autorisations utilisateur, consultez Attribuer et gérer les utilisateurs d'EMR Studio.

Afficher les détails de Studio

New console

Pour afficher les informations relatives à un EMR Studio avec la nouvelle console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR Studio dans le menu de navigation de gauche, choisissez Studios.
3. Sélectionnez le Studio dans la liste Studios pour ouvrir la page détaillée du Studio. La page détaillée du Studio inclut des informations sur les Paramètres du studio, telles que la Description, le VPC et les Sous-réseaux du Studio.

Old console

Pour afficher les informations relatives à un EMR Studio avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/home.
2. Dans le volet de navigation de gauche, choisissez EMR Studio.
3. Sélectionnez le Studio dans la liste Studios pour ouvrir la page détaillée du Studio. La page détaillée du Studio inclut des informations sur les Paramètres du studio, telles que la Description, le VPC et les Sous-réseaux du Studio.

CLI

Pour récupérer les informations d'un EMR Studio par ID de Studio à l'aide de l'AWS CLI

Utilisez la commande `describe-studio` suivante de l'AWS CLI pour obtenir des informations détaillées sur un EMR Studio spécifique. Pour plus d'informations, consultez la référence de la commande AWS CLI.

```
aws emr describe-studio \
```

```
  --studio-id <id-of-studio-to-describe> \
```

Pour récupérer une liste des EMR Studios à l'aide de l'AWS CLI

Utilisez la commande `list-studios` suivante de l'AWS CLI. Pour plus d'informations, consultez la référence de la commande AWS CLI.

```
aws emr list-studios
```

Voici un exemple de valeur de retour pour la commande `list-studios` au format JSON.

```
{
    "Studios": [
        {
            "AuthMode": "IAM",
            "VpcId": "vpc-b21XXXXX",
            "Name": "example-studio-name",
            "Url": "https://es-7HWP74SNGDXXXXXXXXXXXXXX.emrstudio-prod.us-
east-1.amazonaws.com",
            "CreationTime": 1605672582.781,
            "StudioId": "es-7HWP74SNGDXXXXXXXXXXXXXX",
            "Description": "example studio description"
        }
    ]
}
```

Surveiller les actions Amazon EMR Studio

Afficher l'activité de l'EMR Studio et de l'API

EMR Studio est intégré à AWS CloudTrail, un service qui enregistre les actions effectuées par un utilisateur, un rôle ou un service IAM ou par un autre service AWS dans EMR Studio. CloudTrail capture les appels d'API pour EMR Studio sous forme d'événements. Vous pouvez consulter les événements à l'aide de la CloudTrail console à l'adresse https://console.aws.amazon.com/cloudtrail/.

Les événements EMR Studio fournissent des informations telles que l'utilisateur Studio ou IAM qui fait une requête et le type de requête.

> **Note**
>
> Les actions sur le cluster, telles que l'exécution de tâches de bloc-notes, n'émettent pas AWS CloudTrail.

Vous pouvez également créer un journal pour la diffusion continue des CloudTrail événements EMR Studio vers un compartiment Amazon S3. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.

Exemple CloudTrail d'événement : un utilisateur appelle l' DescribeStudioAPI

Voici un exemple d'AWS CloudTrailévénement créé lorsqu'un utilisateur appelle l'DescribeStudioAPI. `admin` CloudTrail enregistre le nom d'utilisateur sous la forme`admin`.

> **Note**
>
> Pour protéger les informations de Studio, l'événement d'API EMR Studio pour DescribeStudio exclut une valeur pour. `responseElements`

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDXXXXXXXXXXXXXXXXX",
      "arn":"arn:aws:iam::653XXXXXXXXX:user/admin",
      "accountId":"653XXXXXXXXX",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"admin"
   },
   "eventTime":"2021-01-07T19:13:58Z",
   "eventSource":"elasticmapreduce.amazonaws.com",
   "eventName":"DescribeStudio",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"72.XX.XXX.XX",
   "userAgent":"aws-cli/1.18.188 Python/3.8.5 Darwin/18.7.0 botocore/1.19.28",
   "requestParameters":{
      "studioId":"es-905XXXXXXXXXXXXXXXXXXXXX"
   },
   "responseElements":null,
```

```
   "requestID":"0fxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "eventID":"b0xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "readOnly":true,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "eventCategory":"Management",
   "recipientAccountId":"653XXXXXXXXX"
}
```

Afficher l'activité des utilisateurs et des tâches Spark

Pour consulter l'activité des tâches Spark par les utilisateurs d'Amazon EMR Studio, vous pouvez configurer l'emprunt d'identité de l'utilisateur sur un cluster. Avec l'emprunt d'identité de l'utilisateur, chaque tâche Spark soumise depuis un espace de travail est associée à l'utilisateur de Studio qui a exécuté le code.

Lorsque l'emprunt d'identité de l'utilisateur est activé, Amazon EMR crée un répertoire d'utilisateurs HDFS sur le nœud primaire du cluster pour chaque utilisateur qui exécute du code dans l'espace de travail. Par exemple, si l'utilisateur `studio-user-1@example.com` exécute du code, vous pouvez vous connecter au nœud primaire et voir que `hadoop fs -ls /user` a un répertoire pour `studio-user-1@example.com`.

Pour configurer l'emprunt d'identité de l'utilisateur Spark, définissez les propriétés suivantes dans les classifications de configuration suivantes :

- `core-site`
- `livy-conf`

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

```
        }
    }
]
```

Pour voir les pages du serveur d'historique, consultez Déboguer des applications et des tâches avec EMR Studio. Vous pouvez également vous connecter au nœud primaire du cluster à l'aide de SSH pour afficher les interfaces web des applications. Pour plus d'informations, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Mettre à jour un Amazon EMR Studio

Après avoir créé un EMR Studio, vous pouvez mettre à jour les attributs suivants à l'aide de l'AWS CLI :

- Name (Nom)
- Description
- Emplacement S3 par défaut
- Sous-réseaux

Pour mettre à jour un EMR Studio à l'aide de l'AWS CLI

Utilisez la commande `update-studio` de l'AWS CLI pour mettre à jour un EMR Studio. Pour plus d'informations, consultez la référence de la commande AWS CLI.

> **Note**
>
> Vous pouvez associer un Studio à un maximum de 5 sous-réseaux. Ces sous-réseaux doivent appartenir au même VPC que le Studio. La liste des ID de sous-réseau que vous soumettez à la commande `update-studio` peut inclure de nouveaux ID de sous-réseau, mais doit également inclure tous les ID de sous-réseau que vous avez déjà associés au Studio. Vous ne pouvez pas supprimer de sous-réseaux d'un Studio.

```
aws emr update-studio \
 --studio-id <example-studio-id-to-update> \
 --name <example-new-studio-name> \
 --subnet-ids <old-subnet-id-1 old-subnet-id-2 old-subnet-id-3 new-subnet-id> \
```

Pour vérifier les modifications, utilisez la commande `describe-studio` de l'AWS CLI et spécifiez votre ID de Studio. Pour plus d'informations, consultez la référence de la commande AWS CLI.

```
aws emr describe-studio \
 --studio-id <id-of-updated-studio> \
```

Supprimer un Amazon EMR Studio et des espaces de travail

Lorsque vous supprimez un Studio, EMR Studio supprime toutes les attributions d'utilisateurs et de groupes IAM Identity Center associées au Studio.

> **Note**
>
> Lorsque vous supprimez un Studio, Amazon EMR ne supprime pas les espaces de travail associés à ce Studio. Vous devez supprimer les espaces de travail de votre Studio séparément.

Supprimer des espaces de travail

Console

Chaque espace de travail EMR Studio étant une instance de bloc-notes EMR, vous pouvez utiliser la console de gestion Amazon EMR pour supprimer des espaces de travail. Vous pouvez supprimer des espaces de travail à l'aide de la console Amazon EMR avant ou après avoir supprimé votre Studio.

Pour supprimer un espace de travail à l'aide de la console Amazon EMR

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Blocs-notes.
3. Sélectionnez le ou les espaces de travail que vous souhaitez supprimer.
4. Choisissez Supprimer, puis à nouveau Supprimer pour confirmer.
5. Suivez les instructions relatives à la suppression d'objets dans le Guide de l'utilisateur de la console Amazon Simple Storage Service si vous souhaitez supprimer les fichiers de bloc-notes associés à l'espace de travail supprimé d'Amazon S3.

EMR Studio UI

From the Workspace UI

Supprimez un Workspace et les fichiers de sauvegarde rattachés dans EMR Studio

1. Connectez-vous à votre EMR Studio à l'aide de votre URL d'accès au studio et choisissez Espaces de travail dans le menu de navigation de gauche.
2. Trouvez votre Workspace dans la liste et cochez la case à côté de son nom. Vous pouvez sélectionner plusieurs Workspaces à supprimer en même temps.
3. Choisissez Supprimer dans le coin supérieur droit de la liste Espaces de travail et confirmez que vous souhaitez supprimer les Workspaces sélectionnés. Choisissez Supprimer pour confirmer.
4. Si vous souhaitez supprimer les fichiers de bloc-notes rattachés au Workspace supprimé d'Amazon S3, suivez les instructions relatives à la suppression d'objets dans le Guide de l'utilisateur de la console Amazon Simple Storage Service. Si vous n'avez pas créé le Studio, consultez votre administrateur afin de déterminer l'emplacement de sauvegarde Amazon S3 pour le Workspace supprimé.

From the Workspaces list

Supprimer un Workspace ainsi que les fichiers de sauvegarde rattachés de la liste des Workspaces

1. Accédez à la liste des Workspaces dans la console.
2. Sélectionnez le Workspace que vous souhaitez supprimer dans la liste, puis choisissez Actions.
3. Sélectionnez Delete (Supprimer).
4. Si vous souhaitez supprimer les fichiers de bloc-notes rattachés au Workspace supprimé d'Amazon S3, suivez les instructions relatives à la suppression d'objets dans le Guide de l'utilisateur de la console Amazon Simple Storage Service. Si vous n'avez pas créé le Studio, consultez votre administrateur afin de déterminer l'emplacement de sauvegarde Amazon S3 pour le Workspace supprimé.

Supprimer un EMR Studio

New console

Pour supprimer un EMR Studio avec la nouvelle console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR Studio dans le menu de navigation de gauche, choisissez Studios.
3. Sélectionnez le Studio dans la liste Studios avec le bouton situé à gauche du nom du Studio. Sélectionnez Delete (Supprimer).

Old console

Pour supprimer un EMR Studio avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/home.
2. Dans le volet de navigation de gauche, choisissez EMR Studio.
3. Sélectionnez le Studio dans la liste Studios et choisissez Supprimer.

CLI

Pour supprimer un EMR Studio avec l'AWS CLI

Utilisez la commande `delete-studio` de l'AWS CLI pour supprimer un EMR Studio. Pour plus d'informations, consultez la référence de la commande AWS CLI.

```
aws emr delete-studio --studio-id <id-of-studio-to-delete>
```

Chiffrement des blocs-notes et des fichiers de l'espace de travail EMR Studio

Dans EMR Studio, vous pouvez créer et configurer différents espaces de travail pour organiser et exécuter des blocs-notes. Ces espaces de travail stockent les blocs-notes et les fichiers associés dans le compartiment Amazon S3 que vous avez spécifié. Par défaut, ces fichiers sont chiffrés avec des clés gérées par Amazon S3 (SSE-S3) avec le chiffrement côté serveur comme niveau de chiffrement de base. Vous pouvez également choisir d'utiliser des clés KMS gérées par le client (SSE-KMS) pour chiffrer vos fichiers. Vous pouvez le faire en utilisant la console de gestion Amazon EMR ou via le AWS SDK AWS CLI et lors de la création d'un studio EMR.

Le chiffrement du stockage de l'espace de travail EMR Studio est disponible dans toutes les régions où EMR Studio est disponible.

Prérequis

Avant de pouvoir chiffrer le bloc-notes et les fichiers de l'espace de travail EMR Studio, vous AWS Key Management Service devez créer une clé de responsable client (CMK) symétrique dans la Compte AWS même région que votre EMR Studio.

Votre politique de ressources AWS KMS doit disposer des autorisations d'accès nécessaires pour le rôle de service de votre EMR Studio. Voici un exemple de politique IAM octroyant des autorisations d'accès minimales pour le chiffrement du stockage EMR Studio Workspace :

```
{
    "Sid": "AllowEMRStudioServiceRoleAccess",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::<ACCOUNT_ID>:role/<ROLE_NAME>"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:CallerAccount": "<ACCOUNT_ID>",
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::<S3_BUCKET_NAME>",
            "kms:ViaService": "s3.<AWS_REGION>.amazonaws.com"
        }
    }
}
```

Votre rôle de service EMR Studio doit également disposer des autorisations d'accès nécessaires pour utiliser votre AWS KMS clé. Voici un exemple de politique IAM octroyant les autorisations d'accès minimales pour le chiffrement du stockage EMR Studio Workspace :

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEMRStudioWorkspaceStorageEncryptionAccess",
            "Effect": "Allow",
            "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey",
               "kms:ReEncryptFrom",
               "kms:ReEncryptTo",
               "kms:DescribeKey"
            ],
            "Resource": ["arn:aws:kms:<REGION>:<ACCOUNT_ID>:key/<KEY_IDENTIFIER>"]
        }
    ]
}
```

Configuration

Procédez comme suit pour créer un nouveau studio EMR qui utilise le chiffrement du stockage de l'espace de travail.

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Studios, puis Create Studio.
3. Pour l'emplacement S3 pour le stockage, entrez ou choisissez un chemin Amazon S3. Il s'agit de l'emplacement Amazon S3 où Amazon EMR stocke les blocs-notes et les fichiers de l'espace de travail.
4. Pour Rôle de service, entrez ou choisissez un rôle IAM. Il s'agit du rôle IAM assumé par Amazon EMR.
5. Choisissez Chiffrer les fichiers de l'espace de travail avec votre propre AWS KMS clé.
6. Entrez ou choisissez une AWS KMS clé à utiliser pour chiffrer les blocs-notes et les fichiers de l'espace de travail dans Amazon S3.
7. Choisissez Create Studio ou Create Studio et lancez Workspaces.
8. Choisissez Chiffrer les fichiers de l'espace de travail avec votre propre AWS KMS clé.
9. Entrez ou choisissez une option AWS KMS à utiliser pour chiffrer les blocs-notes et les fichiers de l'espace de travail dans Amazon S3.
10. Choisissez Save Changes (Enregistrer les modifications).

Les étapes suivantes montrent comment mettre à jour un EMR Studio et configurer le chiffrement du stockage de l'espace de travail.

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez un studio EMR existant dans la liste, puis choisissez Modifier.
3. Choisissez Chiffrer les fichiers de l'espace de travail avec votre propre AWS KMS clé.
4. Entrez ou choisissez une option AWS KMS à utiliser pour chiffrer les blocs-notes et les fichiers de l'espace de travail dans Amazon S3.
5. Choisissez Save Changes (Enregistrer les modifications).

Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio

À propos des groupes de sécurité EMR Studio

Amazon EMR Studio utilise deux groupes de sécurité pour contrôler le trafic réseau entre les espaces de travail du Studio et un cluster Amazon EMR attaché exécuté sur Amazon EC2 :

- Un groupe de sécurité moteur qui utilise le port 18888 pour communiquer avec un cluster Amazon EMR attaché exécuté sur Amazon EC2.
- Un groupe de sécurité d'espace de travail associé aux espaces de travail d'un Studio. Ce groupe de sécurité inclut une règle HTTPS sortante pour permettre à l'espace de travail d'acheminer le trafic vers Internet et doit autoriser le trafic sortant vers Internet sur le port 443 pour permettre de lier les référentiels Git à un espace de travail.

EMR Studio utilise ces groupes de sécurité en plus des groupes de sécurité associés à un cluster EMR attaché à un espace de travail.

Vous devez créer ces groupes de sécurité lorsque vous utilisez l'AWS CLI pour créer un Studio.

> **Note**
>
> Vous pouvez personnaliser les groupes de sécurité pour EMR Studio à l'aide de règles adaptées à votre environnement, mais vous devez inclure les règles indiquées sur cette page. Votre groupe de sécurité d'espace de travail ne peut autoriser aucun trafic entrant, et le groupe de sécurité moteur doit autoriser le trafic entrant en provenance du groupe de sécurité d'espace de travail.

Utiliser les groupes de sécurité EMR Studio par défaut

Lorsque vous utilisez la console Amazon EMR, vous pouvez choisir les groupes de sécurité par défaut suivants. Les groupes de sécurité par défaut sont créés par EMR Studio en votre nom et incluent les règles d'entrée et de sortie minimales requises pour les espaces de travail d'un EMR Studio.

- `DefaultEngineSecurityGroup`
- `DefaultWorkspaceSecurityGroupGit` ou `DefaultWorkspaceSecurityGroupWithoutGit`

Prérequis

Pour créer les groupes de sécurité pour EMR Studio, vous devez disposer d'un cloud privé virtuel (VPC) Amazon pour l'EMR Studio. Vous choisissez ce VPC lorsque vous créez les groupes de sécurité. Il doit s'agir du VPC que vous spécifiez lorsque vous créez le Studio. Si vous prévoyez d'utiliser Amazon EMR sur EKS avec EMR Studio, choisissez le VPC de vos composants master de cluster Amazon EKS.

Instructions

Suivez les instructions de la rubrique Création d'un groupe de sécurité du Guide de l'utilisateur Amazon EC2 pour les instances Linux afin de créer un groupe de sécurité moteur et un groupe de sécurité d'espace de travail dans votre VPC. Les groupes de sécurité doivent inclure les règles résumées dans les tableaux suivants.

Lorsque vous créez des groupes de sécurité pour EMR Studio, notez les ID des deux. Vous spécifiez chaque groupe de sécurité par ID lorsque vous créez un Studio.

Groupe de sécurité moteur

EMR Studio utilise le port 18888 pour communiquer avec un cluster attaché.

Règles entrantes

Type	Protocole	Port	Destination	Description
TCP	TCP	18888	Votre groupe de sécurité	Autorisez le trafic provenant de toutes les ressources du groupe

Type	Protocole	Port	Destination	Description
			EMR Studio Workspace.	de sécurité Workspace pour EMR Studio.

Groupe de sécurité Workspace

Ce groupe de sécurité est rattaché aux Workspaces d'un EMR Studio.

Règles sortantes

Type	Protocole	Port	Destination	Description
TCP	TCP	18888	Votre groupe de sécurité moteur EMR Studio.	Autorisez le trafic provenant de toute ressource du groupe de sécurité moteur pour EMR Studio.
HTTPS	TCP	443	0.0.0.0/0	Autorisez le trafic vers Internet à relier les référentiels Git hébergés publiquem ent aux Workspaces.

Créer des modèles AWS CloudFormation pour Amazon EMR Studio

À propos des modèles de cluster EMR Studio

Vous pouvez créer des AWS CloudFormation modèles pour aider les utilisateurs d'EMR Studio à lancer de nouveaux clusters Amazon EMR dans un espace de travail. CloudFormation les modèles sont des fichiers texte formatés en JSON ou YAML. Dans un modèle, vous décrivez une pile de AWS ressources et expliquez CloudFormation comment les mettre à votre disposition. Pour EMR Studio, vous pouvez créer un ou plusieurs modèles décrivant un cluster Amazon EMR Studio.

Vous organisez vos modèles dans AWS Service Catalog. AWS Service Catalog vous permet de créer et de gérer des services informatiques couramment déployés appelés produits sur AWS. Vous collectez vos modèles sous forme de produits dans un portefeuille que vous partagez avec

les utilisateurs de EMR Studio. Après avoir créé des modèles de cluster, les utilisateurs de Studio peuvent lancer un nouveau cluster pour un Workspace avec l'un de vos modèles. Les utilisateurs doivent être autorisés à créer de nouveaux clusters à partir de modèles. Vous pouvez définir les autorisations des utilisateurs dans les politiques d'autorisation de votre EMR Studio.

Pour en savoir plus sur les CloudFormation modèles, consultez la section Modèles du guide de AWS CloudFormation l'utilisateur. Pour plus d'informations sur AWS Service Catalog, consultez Qu'est-ce qu'un AWS Service Catalog.

La vidéo suivante montre comment configurer des modèles de cluster dans AWS Service Catalog pour EMR Studio. Pour en savoir plus, consultez l'article de blog Créer un environnement en libre-service pour chaque secteur d'activité à l'aide d'Amazon EMR et de Service Catalog.

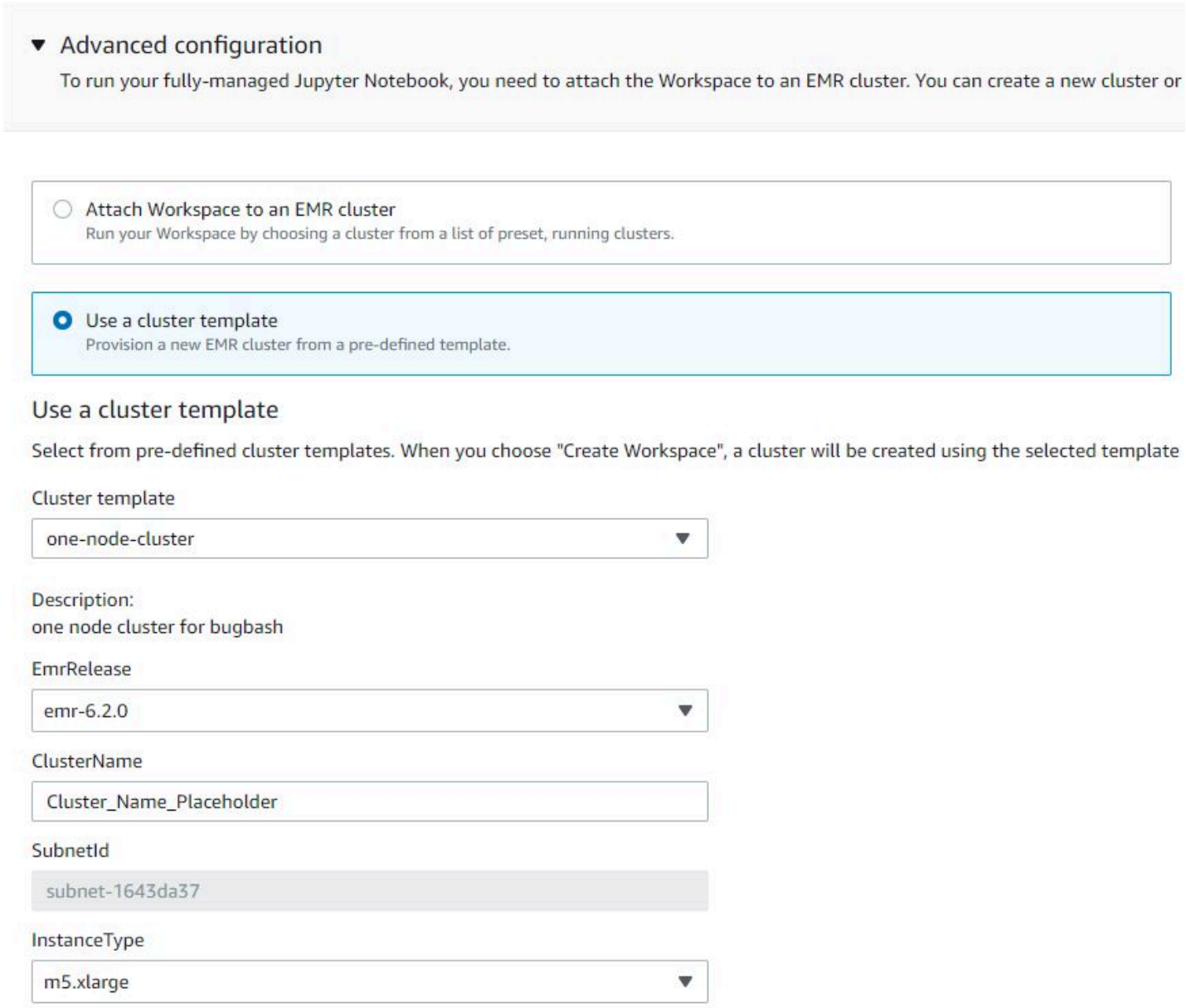
Paramètres de modèle facultatifs

Vous pouvez inclure des options supplémentaires dans la section `Parameters` de votre modèle. Les paramètres permettent aux utilisateurs Studio de saisir ou de sélectionner des valeurs personnalisées pour un cluster. Par exemple, vous pouvez ajouter un paramètre qui permet aux utilisateurs de sélectionner une version Amazon EMR en particulier. Pour plus d'informations, consultez Paramètres dans le Guide de l'utilisateur AWS CloudFormation.

La section `Parameters` d'exemple suivante définit des paramètres d'entrée supplémentaires tels que `ClusterName`, version `EmrRelease` et `ClusterInstanceType`.

```
Parameters:
  ClusterName:
    Type: "String"
    Default: "Cluster_Name_Placeholder"
  EmrRelease:
    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"
```

Lorsque vous ajoutez des paramètres, les utilisateurs de Studio voient des options de formulaire supplémentaires après avoir sélectionné un modèle de cluster. L'image suivante montre des options de formulaire supplémentaires pour les EmrReleaseversions ClusterName, et InstanceType.



Prérequis

Avant de créer un modèle de cluster, assurez-vous que vous disposez des autorisations IAM pour accéder à la console d'administration de Service Catalog. Vous devez également disposer des autorisations IAM suffisantes pour effectuer les tâches administratives de Service Catalog. Pour plus d'informations, consultez Octroi d'autorisations aux administrateurs de Service Catalog.

Instructions

Créer des modèles de clusters EMR à l'aide de Service Catalog

1. Créez un ou plusieurs CloudFormation modèles. C'est à vous de décider où vous stockez vos modèles. Les modèles étant des fichiers texte formatés, vous pouvez les charger sur Amazon S3 ou les conserver dans votre système de fichiers local. Pour en savoir plus sur les CloudFormation modèles, consultez la section Modèles du guide de AWS CloudFormation l'utilisateur.

   Utilisez les règles suivantes pour nommer vos modèles ou vérifiez vos noms par rapport au modèle `[a-zA-Z0-9][a-zA-Z0-9._-]*`.

   - Le nom des modèles doit commencer par un chiffre ou une lettre.
   - Les noms des modèles ne peuvent être composés que de lettres, de chiffres, de points (.), de traits de soulignement (_) et de tirets (-).

   Chaque modèle de cluster que vous créez doit inclure les options suivantes :

   Paramètres d'entrée

   - ClusterName — Un nom pour le cluster afin d'aider les utilisateurs à l'identifier une fois qu'il a été provisionné.

   Sortie

   - `ClusterId` : l'ID du cluster EMR nouvellement alloué.

   Voici un exemple de modèle AWS CloudFormation au format YAML pour un cluster à deux nœuds. L'exemple de modèle inclut les options de modèle requises et définit des paramètres d'entrée supplémentaires pour `EmrRelease` et `ClusterInstanceType`.

```
awsTemplateFormatVersion: 2010-09-09

Parameters:
  ClusterName:
    Type: "String"
    Default: "Example_Two_Node_Cluster"
  EmrRelease:
```

```
    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"

Resources:
  EmrCluster:
    Type: AWS::EMR::Cluster
    Properties:
      Applications:
      - Name: Spark
      - Name: Livy
      - Name: JupyterEnterpriseGateway
      - Name: Hive
      EbsRootVolumeSize: '10'
      Name: !Ref ClusterName
      JobFlowRole: EMR_EC2_DefaultRole
      ServiceRole: EMR_DefaultRole_V2
      ReleaseLabel: !Ref EmrRelease
      VisibleToAllUsers: true
      LogUri:
        Fn::Sub: 's3://aws-logs-${AWS::AccountId}-${AWS::Region}/elasticmapreduce/'
      Instances:
        TerminationProtected: false
        Ec2SubnetId: 'subnet-ab12345c'
        MasterInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
        CoreInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
          Market: ON_DEMAND
          Name: Core

Outputs:
  ClusterId:
    Value:
```

```
      Ref: EmrCluster
    Description: The ID of the  EMR cluster
```

2. Créez un portefeuille pour vos modèles de cluster dans le même compte AWS que votre studio.

   a. Ouvrez la AWS Service Catalog console à l'adresse https://console.aws.amazon.com/servicecatalog/.

   b. Dans le menu de navigation de gauche, choisissez Portefeuilles.

   c. Entrez les informations demandées sur la page Créer un portefeuille.

   d. Choisissez Créer. AWS Service Catalog crée le portefeuille et affiche les détails du portefeuille.

3. Suivez les étapes ci-dessous pour ajouter vos modèles de cluster en tant que produits AWS Service Catalog.

   a. Accédez à la page Produits sous Administration dans la console de gestion AWS Service Catalog.

   b. Choisissez Importer un nouveau produit.

   c. Entrez le nom du produit et le propriétaire.

   d. Spécifiez votre fichier modèle sous Détails de la version.

   e. Choisissez Réviser pour vérifier les paramètres de votre produit, puis Créer un produit.

4. Suivez les étapes ci-dessous pour ajouter vos produits à votre portefeuille.

   a. Accédez à la page Produits dans la console de gestion AWS Service Catalog.

   b. Choisissez votre produit, sélectionnez Actions, puis choisissez Ajouter un produit au portefeuille.

   c. Choisissez votre portefeuille, puis choisissez Ajouter un produit au portefeuille.

5. Créez une contrainte de lancement pour vos produits. Une contrainte de lancement est un rôle IAM qui spécifie les autorisations des utilisateurs pour le lancement d'un produit. Vous pouvez adapter vos contraintes de lancement, mais vous devez autoriser les autorisations d'utilisation CloudFormation, Amazon EMR et. AWS Service Catalog Pour plus d'informations et d'instructions, consultez Contraintes de lancement de Service Catalog.

6. Appliquez votre contrainte de lancement à chaque produit de votre portefeuille. Vous devez appliquer la contrainte de lancement à chaque produit individuellement.

   a. Sélectionnez votre portefeuille sur la page Portefeuilles de la console de gestion AWS Service Catalog.

b. Cliquez sur l'onglet Constraints (Contraintes) puis sur Create constraint (Créer une contrainte).

c. Sélectionnez votre produit puis choisissez Produit sous Type de contrainte. Choisissez Continuer.

d. Sélectionnez votre rôle de contrainte de lancement dans la section Contrainte de lancement, puis choisissez Créer.

7. Accordez l'accès à votre portefeuille.

a. Sélectionnez votre portefeuille sur la page Portefeuilles de la console de gestion AWS Service Catalog.

b. Développez l'onglet Groupes, rôles et utilisateurs et choisissez Ajouter des groupes, des rôles, des utilisateurs.

c. Recherchez votre rôle IAM dans EMR Studio dans l'onglet Rôles, sélectionnez votre rôle, puis choisissez Ajouter un accès.

Si vous utilisez...	Octroi de l'accès à…
Authentification IAM	Vos utilisateurs natifs
Fédération IAM	Votre rôle IAM au sein de la fédération
Fédération Identity Center IAM	Votre rôle d'utilisateur EMR Studio

Établissez l'accès et les autorisations pour les référentiels Git

EMR Studio prend en charge les services Git suivants :

- AWS CodeCommit
- GitHub
- Bitbucket
- GitLab

Pour permettre aux utilisateurs d'EMR Studio d'rattacher un référentiel Git à un Workspace, définissez les exigences d'accès et d'autorisation suivantes. Vous pouvez également configurer des

référentiels Git que vous hébergez sur un réseau privé en suivant les instructions de Configurer un référentiel Git hébergé sur un serveur privé pour EMR Studio.

Accès Internet en cluster

Les clusters Amazon EMR exécutés sur Amazon EC2 et les clusters Amazon EMR sur EKS connectés aux Workspaces Studio doivent se trouver dans un sous-réseau privé qui utilise une passerelle de traduction d'adresses réseau (NAT), ou ils doivent être en mesure d'accéder à Internet via une passerelle privée virtuelle. Pour plus d'informations, consultez Options d'Amazon VPC.

Les groupes de sécurité que vous utilisez avec EMR Studio doivent également inclure une règle sortante qui autorise les Workspaces à acheminer le trafic vers Internet à partir d'un cluster EMR rattaché. Pour plus d'informations, consultez Définir des groupes de sécurité pour contrôler le trafic réseau d'EMR Studio.

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

Autorisations pour AWS Secrets Manager

Pour permettre aux utilisateurs d'EMR Studio d'accéder aux référentiels Git contenant des secrets contenus dans AWS Secrets Manager, ajoutez une politique d'autorisation au rôle de service pour EMR Studio qui autorise l'opération `secretsmanager:GetSecretValue`.

Pour plus d'informations sur la façon de lier des référentiels Git à des Workspaces, consultez. Lier des référentiels Git à un Workspace EMR Studio

Configurer un référentiel Git hébergé sur un serveur privé pour EMR Studio

Suivez les instructions ci-dessous pour configurer les référentiels hébergés sur un serveur privé pour Amazon EMR Studio. Fournissez un fichier de configuration contenant des informations sur vos serveurs DNS et Git. EMR Studio utilise ces informations pour configurer des Workspaces capables d'acheminer le trafic vers vos référentiels autogérés.

> **Note**
>
> Si vous configurez `DnsServerIpV4`, EMR Studio utilise votre serveur DNS pour résoudre à la fois votre `GitServerDnsName` et votre point de terminaison Amazon EMR, par exemple `elasticmapreduce.us-east-1.amazonaws.com`. Pour configurer un point de terminaison pour Amazon EMR, connectez-vous à votre point de terminaison via le VPC que vous utilisez avec votre studio. Cela garantit que le point de terminaison Amazon EMR est résolu par défaut en une adresse IP privée. Pour plus d'informations, consultez Connexion à Amazon EMR à l'aide d'un point de terminaison d'un VPC d'interface.

Prérequis

Avant de configurer un référentiel Git hébergé sur un serveur privé pour EMR Studio, vous avez besoin d'un emplacement de stockage Amazon S3 dans lequel EMR Studio peut sauvegarder les Workspaces et les fichiers de bloc-notes Studio. Utilisez le même compartiment S3 que celui que vous avez spécifié lorsque vous créez un studio.

Configurer un ou de plusieurs référentiels Git hébergés sur un serveur privé pour EMR Studio

1. Créez un fichier de configuration à l'aide du modèle suivant. Incluez les valeurs suivantes pour chaque serveur Git que vous souhaitez spécifier dans votre configuration :

   - **`DnsServerIpV4`** : L'adresse IPv4 de votre serveur DNS. Si vous fournissez des valeurs à la fois pour `DnsServerIpV4` et `GitServerIpV4List`, la valeur de `DnsServerIpV4` sera prioritaire et EMR Studio sera utilisera `DnsServerIpV4` pour résoudre votre `GitServerDnsName`.

     > **Note**
     >
     > Pour utiliser des référentiels Git hébergés sur un serveur privé, votre serveur DNS doit autoriser l'accès entrant depuis EMR Studio. Nous vous recommandons vivement de protéger votre serveur DNS contre tout autre accès non autorisé.

   - **`GitServerDnsName`** : Le nom DNS de votre serveur Git. Par exemple `"git.example.com"`.
   - **`GitServerIpV4List`** : Une liste d'adresses IPv4 appartenant à vos serveurs Git.

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. Enregistrez votre fichier de configuration sous `configuration.json`.
3. Chargez le fichier de configuration dans l'emplacement de stockage Amazon S3 dans un dossier appelé `life-cycle-configuration`. Par exemple, si votre emplacement S3 par défaut est `s3://DOC-EXAMPLE-BUCKET/studios`, votre fichier de configuration doit se trouver dans `s3://DOC-EXAMPLE-BUCKET/studios/`**`life-cycle-configuration/configuration.json`**.

> **Important**
>
> Nous vous conseillons vivement de limiter l'accès à votre dossier `life-cycle-configuration` aux administrateurs de Studio et à votre rôle de service EMR Studio, et de protéger `configuration.json` contre tout accès non autorisé. Pour des instructions, consultez la rubrique Contrôle de l'accès à un compartiment avec des politiques utilisateur ou Bonnes pratiques de sécurité pour Amazon S3.

Pour des instructions sur le chargement, consultez les rubriques Création d'un dossier et Chargement d'objets dans le Guide de l'utilisateur d'Amazon Simple Storage Service. Pour appliquer votre configuration à un Workspace existant, fermez et redémarrez le Workspace après avoir chargé votre fichier de configuration sur Amazon S3.

Optimiser les tâches Spark dans EMR Studio

Lorsque vous exécutez une tâche Spark à l'aide d'EMR Studio, vous pouvez suivre quelques étapes pour optimiser les ressources de votre cluster Amazon EMR.

Prolongez votre session Livy

Si vous utilisez Apache Livy avec Spark sur votre cluster Amazon EMR, nous vous recommandons d'augmenter le délai d'expiration de votre session Livy en effectuant l'une des opérations suivantes :

- Lorsque vous créez un cluster Amazon EMR, définissez cette classification de configuration dans le champ Entrer la configuration.

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
        }
    }
]
```

- Pour un cluster EMR déjà en cours d'exécution, connectez-vous à votre cluster en utilisant `ssh` et définissez la classification de configuration `livy-conf` dans `/etc/livy/conf/livy.conf`.

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
        }
    }
]
```

Vous devrez peut-être redémarrer Livy après avoir modifié la configuration.

- Si vous ne voulez pas que votre session Livy expire, définissez la propriété `livy.server.session.timeout-check` sur `false` dans `/etc/livy/conf/livy.conf`.

Exécuter Spark en mode cluster

En mode cluster, le pilote Spark s'exécute sur un nœud principal plutôt que sur le nœud primaire, ce qui améliore l'utilisation des ressources sur le nœud principal.

Pour exécuter votre application Spark en mode cluster au lieu du mode client par défaut, choisissez le mode Cluster lorsque vous définissez le Mode de déploiement lors de la configuration de votre étape Spark dans votre nouveau cluster Amazon EMR. Pour plus d'informations, consultez Présentation du mode cluster dans la documentation Apache Spark.

Augmenter la mémoire du pilote Spark

Pour augmenter la mémoire du pilote Spark, configurez votre session Spark à l'aide de la commande magique `%%configure` de votre bloc-notes EMR, comme dans l'exemple suivant.

```
%%configure -f
{"driverMemory": "6000M"}
```

Utiliser un Amazon EMR Studio

Cette section contient des rubriques qui vous aident à configurer un Amazon EMR Studio et à interagir avec celui-ci.

La vidéo suivante présente des informations pratiques telles que la création d'un espace de travail et le lancement d'un nouveau cluster Amazon EMR à l'aide d'un modèle de cluster. La vidéo montre également comment parcourir un bloc-notes type.

Cette section inclut les rubriques suivantes pour vous aider à travailler dans un EMR Studio :

- Découvrir les bases de l'espace de Workspace
- Configuration de la collaboration dans Workspace
- Exécuter un Workspace EMR Studio avec un rôle d'exécution
- Exécuter les blocs-notes Workspace par programmation
- Consultation des données à l'aide de SQL Explorer

- Attacher un calcul à un espace de travail EMR Studio
- Lier des référentiels Git à un Workspace EMR Studio
- Utilisation de l'éditeur SQL Amazon Athena dans EMR Studio
- CodeWhisperer Intégration d'Amazon à EMR Studio Workspaces
- Déboguer des applications et des tâches avec EMR Studio
- Installation de noyaux et de bibliothèques dans un Workspace EMR Studio
- Améliorer les noyaux avec des commandes magic
- Utiliser des blocs-notes multilingues avec des noyaux Spark

Découvrir les bases de l'espace de Workspace

Lorsque vous utilisez un EMR Studio, vous pouvez créer et configurer différents Workspaces pour organiser et exécuter des blocs-notes. Cette section traite de la création et de l'utilisation des Workspaces. Pour obtenir une présentation conceptuelle, consultez Espaces de travail à la page Comment fonctionne Amazon EMR Studio.

Cette section inclut les rubriques suivantes pour vous aider à utiliser les Workspaces EMR Studio :

- Créer un Workspace EMR Studio
- Lancer un Workspace
- Comprendre l'interface utilisateur de Workspace
- Découvrez des exemples de bloc-notes
- Enregistrer le contenu Workspace
- Supprimer un Workspace et des fichiers de bloc-notes
- Comprendre l'état de Workspace
- Résoudre les problèmes de connectivité dans Workspace

Créer un Workspace EMR Studio

Vous pouvez créer des Workspaces EMR Studio pour exécuter du code de bloc-notes à l'aide de l'interface EMR Studio.

Créer un Workspace dans un EMR Studio

1. Connectez-vous à votre EMR Studio.

2. Choisissez Créer un Workspace.
3. Saisissez un nom d'Workspace et une description. Le fait de nommer un Workspace vous permet de l'identifier sur la page Workspaces.
4. Si vous souhaitez travailler avec d'autres utilisateurs de Studio dans cet Workspace en temps réel, activez la collaboration dans le Workspace. Vous pouvez configurer les collaborateurs après avoir lancé le Workspace.
5. Si vous souhaitez rattacher un cluster à un Workspace, développez la section Configuration avancée. Vous pouvez rattacher un cluster ultérieurement, si vous le souhaitez. Pour plus d'informations, consultez Attacher un calcul à un espace de travail EMR Studio.

   > **Note**
   >
   > Pour configurer un nouveau cluster, vous devez obtenir des autorisations d'accès de la part de votre administrateur.

   Choisissez l'une des options de cluster pour le Workspace et rattachez le cluster. Pour plus d'informations sur la mise en service d'un cluster lors de la création d'un Workspace, consultez Création et rattachement d'un nouveau cluster EMR à un Workspace EMR Studio.
6. Choisissez Créer un Workspace dans le coin inférieur droit de la page.

Après avoir créé un Workspace, EMR Studio ouvre la page Workspaces. Une bannière verte s'affichera en haut de la page, elle indique la réussite de l'opération et vous trouverez le Workspace nouvellement créé dans la liste.

Par défaut, un Workspace est partagé et peut être vu par tous les utilisateurs de Studio. Toutefois, un seul utilisateur peut ouvrir et travailler dans un Workspace à la fois. Pour travailler simultanément avec d'autres utilisateurs, vous pouvez Configuration de la collaboration dans Workspace

Lancer un Workspace

Pour commencer à travailler avec des fichiers de bloc-notes, lancez un Workspace et accédez à l'éditeur de bloc-notes. La page Espaces de travail d'un studio répertorie tous les Workspaces auxquels vous avez accès avec des détails tels que le nom, le statut, l'heure de création et la date de dernière modification.

> **Note**
>
> Si vous disposiez de blocs-notes EMR dans l'ancienne console Amazon EMR, vous pouvez les retrouver dans la nouvelle console en tant qu'Workspaces EMR Studio. Pour accéder aux Workspaces ou en créer, les utilisateurs de EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Si vous avez récemment créé un bloc-notes dans l'ancienne console, vous devrez peut-être actualiser la liste des Workspaces pour l'afficher dans la nouvelle console. Pour plus d'informations sur la transition, consultez Les notebooks Amazon EMR sont disponibles sous forme d'espaces de travail Amazon EMR Studio dans la console et Console Amazon EMR

Lancer un Workspace qui permet de modifier et d'exécuter des blocs-notes

1. Sur la page Workspaces de votre studio, recherchez le Workspace. Vous pouvez filtrer la liste par mot clé ou par valeur de colonne.
2. Choisissez le nom du Workspace pour le lancer dans un nouvel onglet de navigateur. L'ouverture du Workspace peut prendre quelques minutes s'il est inactif. Vous pouvez également sélectionner la ligne du Workspace, puis sélectionner Lancer le Workspace. Choisissez parmi les options de lancement suivantes :

   - Lancement rapide : lancez rapidement votre Workspace avec les options par défaut. Choisissez Lancement rapide si vous souhaitez associer des clusters à l'espace de travail dans JupyterLab.
   - Lancer avec options : lancez votre Workspace avec des options personnalisées. Vous pouvez choisir de le lancer dans Jupyter ou JupyterLab d'associer votre espace de travail à un cluster EMR et de sélectionner vos groupes de sécurité.

   > **Note**
   >
   > Un seul utilisateur peut ouvrir et travailler dans un Workspace à la fois. Si vous sélectionnez un Workspace déjà utilisé, EMR Studio affichera une notification lorsque vous essayez de l'ouvrir. La colonne Utilisateur de la page Workspaces indique l'utilisateur travaillant dans le Workspace.

Comprendre l'interface utilisateur de Workspace

L'interface utilisateur d'EMR Studio Workspace est basée sur l'JupyterLabinterface avec des onglets désignés par des icônes dans la barre latérale gauche. Lorsque vous passez au-dessus d'une icône, vous pouvez voir une infobulle indiquant le nom de l'onglet. Choisissez des onglets dans la barre latérale gauche pour accéder aux panneaux suivants.

- Navigateur de fichiers : affiche les fichiers et les répertoires du Workspace, ainsi que les fichiers et les répertoires des référentiels Git liés.
- Noyaux et terminaux en cours d'exécution : répertorie tous les noyaux et terminaux exécutés dans le Workspace. Pour plus d'informations, consultez la section Gestion des noyaux et des terminaux dans la JupyterLab documentation officielle.
- Git : fournit une interface utilisateur graphique permettant d'exécuter des commandes dans les référentiels Git rattachés au Workspace. Ce panneau est une JupyterLab extension appelée jupyterlab-git. Pour plus d'informations, consultez jupyterlab-git.
- Clusters EMR : associez un cluster au Workspace ou détacher-en un pour exécuter le code du bloc-notes. Le panneau de configuration du cluster EMR fournit également des options de configuration avancées pour vous aider à créer et à rattacher un nouveau cluster au Workspace. Pour plus d'informations, consultez Création et rattachement d'un nouveau cluster EMR à un Workspace EMR Studio.
- Référentiel Git Amazon EMR : vous permet de relier le Workspace à trois référentiels Git au maximum. Pour plus de détails, consultez Lier des référentiels Git à un Workspace EMR Studio.
- Exemples de blocs-notes : fournit une liste d'exemples de blocs-notes que vous pouvez enregistrer dans le Workspace. Vous pouvez également accéder aux exemples en choisissant Exemples de bloc-notes sur la page Lanceur du Workspace.
- Commandes : permet de rechercher et d'exécuter des commandes à l'aide d'un clavier. JupyterLab Pour plus d'informations, consultez la page de la palette de commandes dans la JupyterLab documentation.
- Outils pour bloc-notes : sélectionnez et de définissez des options telles que le type de diapositive de cellule et les métadonnées. L'option Outils de bloc-notes apparaît dans la barre latérale gauche une fois que vous avez ouvert un fichier de bloc-notes.
- Onglets ouverts : répertorie les documents ouverts et les activités dans la zone de travail principale afin que vous puissiez accéder à un onglet ouvert. Pour plus d'informations, consultez la page Onglets et mode document unique dans la JupyterLab documentation.

- Collaboration : vous permet d'activer ou de désactiver la collaboration dans le Workspace et de gérer les collaborateurs. Pour consulter le panneau Collaboration, vous devez disposer des autorisations nécessaires. Pour plus d'informations, consultez Définir la propriété pour la collaboration dans le Workspace.

Découvrez des exemples de bloc-notes

Chaque Workspace EMR Studio inclut un ensemble d'exemples de blocs-notes que vous pouvez utiliser pour explorer les fonctionnalités d'EMR Studio. Pour modifier ou exécuter un exemple de bloc-notes, vous pouvez l'enregistrer dans le Workspace.

Enregistrer un exemple de bloc-notes dans un Workspace

1. Dans la barre latérale gauche, choisissez l'onglet Exemples de bloc-notes pour ouvrir le panneau Exemples de bloc-notes. Vous pouvez également accéder aux exemples en choisissant Exemples de bloc-notes sur la page Lanceur du Workspace.
2. Choisissez un exemple de bloc-notes pour le prévisualiser dans la zone de travail principale. L'exemple est en lecture seule.
3. Pour enregistrer l'exemple de bloc-notes dans le Workspace, choisissez Enregistrer dans le Workspace. EMR Studio enregistre l'exemple dans votre répertoire de base. Après avoir enregistré un exemple de bloc-notes dans le Workspace, vous pouvez le renommer, le modifier et l'exécuter.

Pour plus d'informations sur les exemples de blocs-notes, consultez le référentiel d'exemples GitHub de blocs-notes EMR Studio.

Enregistrer le contenu Workspace

Lorsque vous travaillez dans l'éditeur de bloc-notes d'un Workspace, EMR Studio enregistre pour vous le contenu des cellules du bloc-notes ainsi que le produit dans l'emplacement Amazon S3 rattaché au studio. Ce processus de sauvegarde préserve le travail entre les sessions.

Vous pouvez également enregistrer un bloc-notes en appuyant sur CTRL+S dans l'onglet Ouvrir le bloc-notes, ou en utilisant l'une des options d'enregistrement sous Fichier.

Une autre méthode pour sauvegarder les fichiers du bloc-notes dans un Workspace consiste à rattacher le Workspace à un référentiel Git et à synchroniser vos modifications avec le référentiel distant. Cela vous permet également d'enregistrer et de partager des blocs-notes avec les membres

de l'équipe qui utilisent un autre Workspace ou un autre studio. Pour obtenir des instructions, veuillez consulter Lier des référentiels Git à un Workspace EMR Studio.

Supprimer un Workspace et des fichiers de bloc-notes

Lorsque vous supprimez un fichier de bloc-notes d'un Workspace EMR Studio, vous supprimez le fichier dans le navigateur de fichiers, et EMR Studio supprime sa copie de sauvegarde dans Amazon S3. Vous n'avez aucune autre mesure à prendre pour éviter les frais de stockage lorsque vous supprimez un fichier d'un Workspace.

Lorsque vous supprimez un Workspace complet, ses fichiers et dossiers de bloc-notes restent dans l'emplacement de stockage Amazon S3. Les fichiers continuent d'être soumis à des frais de stockage. Pour éviter les frais de stockage, supprimez tous les fichiers et dossiers sauvegardés rattachés au Workspace que vous avez supprimé d'Amazon S3.

Supprimer un fichier de bloc-notes d'un Workspace EMR Studio

1. Sélectionnez le panneau Navigateur de fichiers dans la barre latérale gauche du Workspace.
2. Sélectionnez le fichier ou le dossier que vous souhaitez supprimer. Cliquez avec le bouton droit de la souris sur la sélection, puis sélectionnez Supprimer. Le fichier disparaît de la liste. EMR Studio supprime le fichier ou le dossier d'Amazon S3 pour vous.

From the Workspace UI

Supprimez un Workspace et les fichiers de sauvegarde rattachés dans EMR Studio

1. Connectez-vous à votre EMR Studio à l'aide de votre URL d'accès au studio et choisissez Espaces de travail dans le menu de navigation de gauche.
2. Trouvez votre Workspace dans la liste et cochez la case à côté de son nom. Vous pouvez sélectionner plusieurs Workspaces à supprimer en même temps.
3. Choisissez Supprimer dans le coin supérieur droit de la liste Espaces de travail et confirmez que vous souhaitez supprimer les Workspaces sélectionnés. Choisissez Supprimer pour confirmer.
4. Si vous souhaitez supprimer les fichiers de bloc-notes rattachés au Workspace supprimé d'Amazon S3, suivez les instructions relatives à la suppression d'objets dans le Guide de l'utilisateur de la console Amazon Simple Storage Service. Si vous n'avez pas créé le Studio, consultez votre administrateur afin de déterminer l'emplacement de sauvegarde Amazon S3 pour le Workspace supprimé.

From the Workspaces list

Supprimer un Workspace ainsi que les fichiers de sauvegarde rattachés de la liste des Workspaces

1. Accédez à la liste des Workspaces dans la console.
2. Sélectionnez le Workspace que vous souhaitez supprimer dans la liste, puis choisissez Actions.
3. Sélectionnez Delete (Supprimer).
4. Si vous souhaitez supprimer les fichiers de bloc-notes rattachés au Workspace supprimé d'Amazon S3, suivez les instructions relatives à la suppression d'objets dans le Guide de l'utilisateur de la console Amazon Simple Storage Service. Si vous n'avez pas créé le Studio, consultez votre administrateur afin de déterminer l'emplacement de sauvegarde Amazon S3 pour le Workspace supprimé.

Comprendre l'état de Workspace

Une fois que vous avez créé un Workspace EMR Studio, il apparaît sous forme de ligne dans la liste Workspaces de votre studio avec son nom, son état, son heure de création et sa date de dernière modification. Le tableau suivant décrit les états d'un Workspace.

État	Description
Démarrage en cours	Le Workspace est en cours de préparation, mais il n'est pas encore prêt à être utilisé. Vous ne pouvez pas ouvrir un Workspace lorsque son état est « Démarrage en cours ».
Prêt	Vous pouvez ouvrir le Workspace pour utiliser l'éditeur de bloc-notes, mais vous devez rattacher le Workspace à un cluster EMR avant de pouvoir exécuter le code du bloc-notes.
Attachement en cours	Le Workspace est en cours de rattachement à un cluster.
Attaché	Le Workspace est rattaché à un cluster EMR et prêt à l'emploi ; vous pouvez écrire et exécuter

État	Description
	du code de bloc-notes. Si l'état d'un Workspace n'est pas Attaché, vous devez l'rattacher à un cluster avant de pouvoir exécuter le code du bloc-notes.
Inactif	Le Workspace s'est arrêté. Pour réactiver un Workspace inactif, sélectionnez-le dans la liste des Workspaces. L'état passe de Inactif à Démarrage puis Prêt lorsque vous sélectionnez le Workspace.
Arrêt en cours	Le Workspace est en train de fermer et sera bientôt Inactif. Lorsque vous arrêtez un Workspace, il met fin à tous les noyaux de bloc-notes correspondants. EMR Studio arrête automatiquement les bloc-notes qui sont inactifs depuis longtemps.
Suppression en cours	Lorsque vous supprimez un Workspace, EMR Studio le marque pour suppression et lance le processus de suppression. Une fois le processus de suppression terminé, le Workspace disparaît de la liste. Lorsque vous supprimez un Workspace, ses fichiers de bloc-notes restent dans l'emplacement de stockage Amazon S3.

Résoudre les problèmes de connectivité dans Workspace

Pour résoudre les problèmes de connectivité d'un Workspace, vous pouvez arrêter et redémarrer ce dernier. Lorsque vous redémarrez un Workspace, EMR Studio lance le Workspace dans une autre zone de disponibilité ou un autre sous-réseau rattaché à votre studio.

Arrêter et redémarrer un Workspace EMR Studio

1. Fermez le Workspace dans le document.

2. Accédez à la liste Espaces de travail dans la console.
3. Sélectionnez votre Workspace dans la liste, puis cliquez sur Actions.
4. Choisissez Arrêter et attendez que l'état du Workspace passe de Arrêt en cours à Inactif.
5. Choisissez à nouveau Actions, puis sélectionnez Démarrer pour redémarrer le Workspace.
6. Attendez que l'état du Workspace passe de Démarrage en cours à Prêt, puis choisissez le nom du Workspace pour le rouvrir dans un nouvel onglet du navigateur.

Configuration de la collaboration dans Workspace

La collaboration dans Workspace vous permet d'écrire et d'exécuter du code de bloc-notes simultanément avec les autres membres de votre équipe. Lorsque vous travaillez dans le même fichier de bloc-notes, vous verrez les modifications apportées par vos collaborateurs. Vous pouvez activer la collaboration au moment de la création du Workspace, ou activer/désactiver la collaborateur plus tard, une fois le Workspace créé.

> **Note**
>
> La collaboration dans un Workspace EMR Studio n’est pas prise en charge avec les applications interactives EMR sans serveur, ni si la propagation d’identité approuvée est activée.

Prérequis

Avant de configurer la collaboration pour un Workspace, assurez-vous d'effectuer les tâches suivantes :

- Assurez-vous que l'administrateur de votre EMR Studio vous a accordé les autorisations nécessaires. Par exemple, l'instruction suivante permet à un utilisateur de configurer la collaboration pour n'importe quel Workspace avec la clé de balise `creatorUserId` dont la valeur correspond à l'ID de l'utilisateur (indiqué par la variable de politique `aws:userId`).

```
{
    "Sid": "UserRolePermissionsForCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
```

```
        "elasticmapreduce:DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
        }
    }
}
```

- Assurez-vous que le rôle de service rattaché à votre EMR Studio dispose des autorisations requises pour activer et configurer la collaboration dans le Workspace, comme dans l'exemple d'instruction suivant.

```
{
    "Sid": "AllowWorkspaceCollaboration",
    "Effect": "Allow",
    "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
    ],
    "Resource": "*"
}
```

  Pour plus d’informations, consultez Créer une fonction du service EMR Studio.

Activer la collaboration dans Workspace et ajouter des collaborateurs

1. Dans votre Workspace, choisissez l'icône Collaboration dans l'écran du lanceur ou en bas du panneau de gauche.

> **Note**
>
> Le panneau Collaboration ne s'affichera que si l'administrateur de votre studio vous a autorisé à configurer la collaboration pour le Workspace. Pour plus d'informations, consultez Définir la propriété pour la collaboration dans le Workspace.

2. Assurez-vous que le bouton Autoriser la collaboration dans Workspace est activé. Lorsque vous activez la collaboration, seuls vous et les collaborateurs que vous ajoutez pouvez voir le Workspace dans la liste de la page Workspaces du studio.
3. Entrez le nom du collaborateur. Votre Workspace peut compter un maximum de cinq collaborateurs, vous y compris. Un collaborateur peut être n'importe quel utilisateur ayant accès à votre EMR Studio. Si vous n'entrez pas de collaborateur, le Workspace sera un Workspace privé auquel vous seul pouvez accéder.

   Le tableau suivant indique les valeurs de collaborateur applicables à saisir en fonction du type d'identité du propriétaire.

   > **Note**
   >
   > Un propriétaire ne peut inviter que des collaborateurs ayant le même type d'identité. Par exemple, un utilisateur ne peut ajouter que d'autres utilisateurs, et un utilisateur IAM Identity Center ne peut ajouter que d'autres utilisateurs IAM Identity Center.

Mode d’authentification	Valeur à saisir pour le nom du collaborateur
Authentification IAM	un nom d'utilisateur. Il s'agit du nom que voit un utilisateur lorsqu'il est connecté à la AWS Management Console.
Fédération IAM	Le nom d'un rôle IAM et un nom de session facultatif. Pour ajouter tous les utilisateurs fédérés qui assument le même rôle IAM, spécifiez le nom d'un rôle IAM pour la fédération.

Mode d’authentification	Valeur à saisir pour le nom du collaborateur
	Ajouter un seul utilisateur en tant que collaborateur, spécifier un rôle et un nom de session. Par exemple, `MyRoleNam e:MySessionName` .
SSO	Un nom d'utilisateur IAM Identity Center tel que `user@example.com.`

4. Choisissez Ajouter. Le collaborateur peut désormais voir le Workspace sur sa page EMR Studio Workspaces et lancer le Workspace pour l'utiliser avec vous en temps réel.

> **Note**
>
> Si vous désactivez la collaboration dans Workspace, le Workspace revient à son état partagé et peut être vu par tous les utilisateurs de Studio. Dans l'état partagé, un seul utilisateur de Studio peut ouvrir le Workspace et y travailler à la fois.

Exécuter un Workspace EMR Studio avec un rôle d'exécution

> **Note**
>
> La fonctionnalité du rôle d'exécution décrite sur cette page s'applique uniquement à Amazon EMR exécuté sur Amazon EC2 et ne fait pas référence à la fonctionnalité du rôle d'exécution dans les applications interactives EMR sans serveur. Pour en savoir plus sur l'utilisation des rôles d'exécution dans EMR sans serveur, consultez la rubrique Job runtime roles dans le Guide de l'utilisateur Amazon EMR sans serveur.

Un rôle d'exécution est un rôle AWS Identity and Access Management (IAM) que vous pouvez spécifier lorsque vous soumettez une tâche ou une requête à un cluster Amazon EMR. La tâche ou la requête que vous soumettez à votre cluster EMR utilise le rôle d'exécution pour accéder à AWS des ressources, telles que des objets dans Amazon S3.

Lorsque vous associez un espace de travail EMR Studio à un cluster EMR qui utilise Amazon EMR 6.11 ou version ultérieure, vous pouvez sélectionner un rôle d'exécution pour la tâche ou la requête que vous soumettez afin qu'elle soit utilisée lors de l'accès aux ressources. AWS Toutefois, si le cluster EMR ne prend pas en charge les rôles d'exécution, le cluster EMR n'assumera pas ce rôle lorsqu'il accède aux ressources. AWS

Avant de pouvoir utiliser un rôle d'exécution dans un Workspace Amazon EMR Studio, un administrateur doit configurer les autorisations utilisateur afin que ce dernier puisse appeler l'API `elasticmapreduce:GetClusterSessionCredentials` sur le rôle d'exécution. Lancez ensuite un nouveau cluster doté d'un rôle d'exécution que vous pouvez utiliser avec votre Workspace Amazon EMR Studio.

Sur cette page

- Configurer les autorisations utilisateur pour le rôle d'exécution
- Lancer un nouveau cluster avec un rôle d'exécution
- Utiliser le cluster EMR avec un rôle d'exécution dans Workspaces
- Considérations

Configurer les autorisations utilisateur pour le rôle d'exécution

Configurez les autorisations utilisateur afin que l'utilisateur de Studio puisse appeler l'API `elasticmapreduce:GetClusterSessionCredentials` sur le rôle d'exécution qu'il souhaite utiliser. Vous devez également configurer the section called “Autorisations utilisateur Studio (EC2, EKS)” avant que l'utilisateur puisse commencer à utiliser Studio.

> **Warning**
>
> Pour accorder cette autorisation, créez une condition basée sur la clé de contexte `elasticmapreduce:ExecutionRoleArn` lorsque vous autorisez un utilisateur à appeler les API `GetClusterSessionCredentials`. Les exemples suivants vous montre comment procéder.

```
{
      "Sid": "AllowSpecificExecRoleArn",
      "Effect": "Allow",
      "Action": [
```

```
            "elasticmapreduce:GetClusterSessionCredentials"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "elasticmapreduce:ExecutionRoleArn": [
                    "arn:aws:iam::111122223333:role/test-emr-demo1",
                    "arn:aws:iam::111122223333:role/test-emr-demo2"
                ]
            }
        }
    }
```

L'exemple suivant montre comment autoriser un principal IAM à utiliser un rôle IAM appelé `test-emr-demo3` en tant que rôle d'exécution. En outre, le titulaire de la politique ne pourra accéder aux clusters Amazon EMR qu'avec l'ID du cluster `j-123456789`.

```
{
    "Sid":"AllowSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": [
          "arn:aws:elasticmapreduce:<region>:111122223333:cluster/j-123456789"
     ],
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo3"
            ]
        }
    }
}
```

L'exemple suivant permet à un principal IAM d'utiliser n'importe quel rôle IAM dont le nom commence par la chaîne `test-emr-demo4` comme rôle d'exécution. En outre, le titulaire de la politique ne pourra accéder qu'aux clusters Amazon EMR étiquetés avec la paire clé-valeur `tagKey:tagValue`.

```
{
    "Sid":"AllowSpecificExecRoleArn",
```

```
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": "*",
    "Condition":{
        "StringEquals":{
             "elasticmapreduce:ResourceTag/tagKey": "tagValue"
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo4*"
            ]
        }
    }
}
```

Lancer un nouveau cluster avec un rôle d'exécution

Maintenant que vous disposez des autorisations requises, lancez un nouveau cluster avec un rôle d'exécution que vous pouvez utiliser avec votre Workspace Amazon EMR Studio.

Si vous avez déjà lancé un nouveau cluster doté d'un rôle d'exécution, vous pouvez passer à la section the section called “Utiliser le cluster avec votre Workspace”.

1. Tout d'abord, remplissez les conditions requises dans la section Rôles d'exécution pour les étapes Amazon EMR.
2. Lancez ensuite un cluster avec les paramètres suivants pour utiliser les rôles d'exécution avec Amazon EMR Studio Workspaces. Pour savoir comment mettre à jour votre cluster, consultez Spécification d'une configuration de sécurité pour un cluster.

   - Pour la version emr-6.11.0 ou ultérieure.
   - Sélectionnez Spark, Livy et Jupyter Enterprise Gateway comme applications de cluster.
   - Utilisez la configuration de sécurité que vous avez créée à l'étape précédente.
   - Vous pouvez éventuellement activer Lake Formation pour votre cluster EMR. Pour plus d’informations, consultez Activation de Lake Formation avec Amazon EMR.

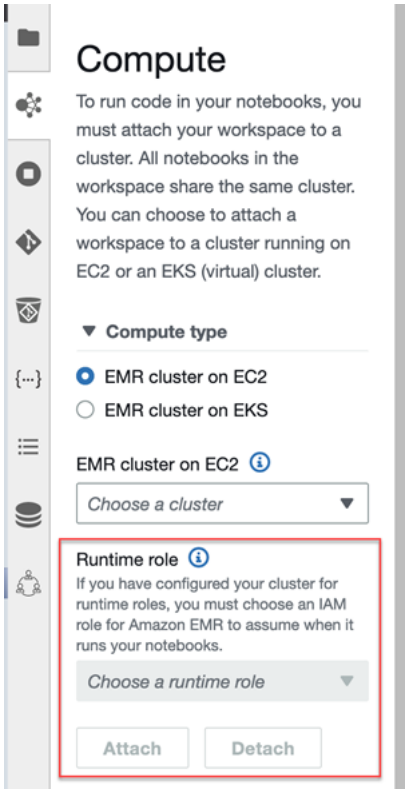
Après avoir lancé votre cluster, vous pouvez utiliser le cluster doté de rôles d'exécution avec un Workspace EMR Studio.

> **Note**
>
> La ExecutionRoleArnvaleur n'est actuellement pas prise en charge par l'opération d' StartNotebookExecutionAPI lorsqu'elle estEMR. `ExecutionEngineConfig.Type`

Utiliser le cluster EMR avec un rôle d'exécution dans Workspaces

Après avoir configuré et lancé votre cluster, vous pouvez utiliser le cluster doté de rôles d'exécution avec un Workspace EMR Studio.

1. Créer un nouvel Workspace ou lancer un Workspace existant. Pour plus d'informations, consultez Créer un Workspace EMR Studio.
2. Choisissez l'onglet Clusters EMR dans la barre latérale gauche de votre Workspace ouvert, développez la section Type de calcul et choisissez votre cluster dans le menu Cluster EMR sur EC2, ainsi que le rôle d'exécution dans le menu Rôle d'exécution.



3. Choisissez Attacher pour rattacher le cluster doté du rôle d'exécution à votre Workspace.

Considérations

Tenez compte des considérations suivantes lorsque vous utilisez un cluster doté de rôles d'exécution avec votre Workspace Amazon EMR Studio :

- Vous ne pouvez sélectionner un rôle d'exécution que lorsque vous rattachez un Workspace EMR Studio à un cluster EMR qui utilise Amazon EMR version 6.11 ou ultérieure.
- La fonctionnalité du rôle d'exécution décrite sur cette page n'est prise en charge qu'avec Amazon EMR exécuté sur Amazon EC2, et n'est pas prise en charge avec les applications interactives EMR sans serveur. Pour en savoir plus sur les rôles d'exécution dans EMR sans serveur, consultez la rubrique Job runtime roles dans le Guide de l'utilisateur Amazon EMR sans serveur.
- Bien que vous deviez configurer des autorisations supplémentaires avant de pouvoir spécifier un rôle d'exécution lorsque vous soumettez une tâche à un cluster, vous n'avez pas besoin d'autorisations supplémentaires pour accéder aux fichiers générés par un Workspace EMR Studio. Les autorisations pour ces fichiers sont les mêmes que celles des fichiers générés à partir de clusters sans rôles d'exécution.
- Vous ne pouvez pas utiliser SQL Explorer dans un Workspace EMR Studio avec un cluster doté d'un rôle d'exécution. Amazon EMR désactive SQL Explorer dans l'interface utilisateur lorsqu'un Workspace est rattaché à un cluster EMR doté de rôles d'exécution.
- Vous ne pouvez pas utiliser le mode Collaboration dans un Workspace EMR Studio avec un cluster doté d'un rôle d'exécution. Amazon EMR désactive les fonctionnalités de collaboration d'un Workspace lorsqu'un Workspace est rattaché à un cluster EMR doté de rôles d'exécution. Le Workspace restera accessible uniquement à l'utilisateur qui l'a rattaché.
- Vous ne pouvez pas utiliser de rôles d’exécution dans un Studio où la propagation d’identité approuvée IAM Identity Center est activée.
- Vous pourriez recevoir un message d'avertissement « La page n'est peut-être pas sûre ! » depuis l'interface utilisateur de Spark pour un cluster doté de rôles d'exécution. Dans ce cas, contournez l'alerte pour rester sur l'interface utilisateur de Spark.

Exécuter les blocs-notes Workspace par programmation

> **Note**
>
> L'exécution par programmation des blocs-notes n'est pas prise en charge par les applications interactives Amazon EMR sans serveur.

Vous pouvez exécuter vos blocs-notes Amazon EMR Studio Workspace par programmation à l'aide d'un script ou sur l'interface AWS CLI. Pour savoir comment exécuter votre bloc-notes par programmation, consultez Exemples de commandes pour l'exécution de blocs-notes EMR par programmation.

Consultation des données à l'aide de SQL Explorer

> **Note**
>
> SQL Explorer pour EMR Studio n’est pas compatible avec les applications interactives Amazon EMR sans serveur ni dans un Studio où la propagation d’identité approuvée IAM Identity Center est activée.

Cette rubrique fournit des informations pour vous aider à démarrer avec SQL Explorer dans Amazon EMR Studio. SQL Explorer est un outil d'une seule page intégré à votre Workspace qui vous aide à comprendre les sources de données du catalogue de données de votre cluster EMR. Vous pouvez utiliser SQL Explorer pour parcourir vos données, exécuter des requêtes SQL pour récupérer des données et télécharger les résultats des requêtes.

SQL Explorer prend en charge Presto. Avant d'utiliser SQL Explorer, assurez-vous que vous disposez d'un cluster qui utilise Amazon EMR version 5.34.0 ou ultérieure ou version 6.4.0 ou ultérieure avec Presto installé. SQL Explorer d'Amazon EMR Studio ne prend pas en charge les clusters Presto que vous avez configurés avec le chiffrement en transit. Cela est dû au fait que Presto s'exécute en mode TLS sur ces clusters.

Parcourez le catalogue de données de votre cluster

SQL Explorer fournit une interface de navigateur de catalogue que vous pouvez utiliser pour explorer et comprendre comment vos données sont organisées. Par exemple, vous pouvez utiliser le navigateur de catalogue de données pour vérifier les noms des tableaux et des colonnes avant d'écrire une requête SQL.

Parcourir votre catalogue de données

1. Ouvrez SQL Explorer dans votre Workspace.
2. Assurez-vous que votre Workspace est rattaché à un cluster EMR exécuté sur EC2 qui utilise Amazon EMR version 6.4.0 ou ultérieure avec Presto installé. Choisissez un cluster existant ou

créez-en un. Pour plus d'informations, consultez Attacher un calcul à un espace de travail EMR Studio.

3. Sélectionnez une base de données dans la liste déroulante pour la parcourir.
4. Développez un tableau dans votre base de données pour voir les noms des colonnes du tableau. Vous pouvez également saisir un mot clé dans la barre de recherche pour filtrer les résultats dans le tableau.

Exécuter une requête SQL pour récupérer des données

Récupérer des données à l'aide d'une requête SQL et télécharger les résultats

1. Ouvrez SQL Explorer dans votre Workspace.
2. Assurez-vous que votre Workspace est connecté à un cluster EMR exécuté sur EC2 avec Presto et Spark installés. Choisissez un cluster existant ou créez-en un. Pour plus d'informations, consultez Attacher un calcul à un espace de travail EMR Studio.
3. Sélectionnez Ouvrir l'éditeur pour ouvrir un nouvel onglet d'éditeur dans votre Workspace.
4. Rédigez votre requête SQL dans l'onglet de l'éditeur.
5. Cliquez sur Exécuter.
6. Consultez les résultats de votre requête sous Aperçu des résultats. SQL Explorer affiche les 100 premiers résultats par défaut. Vous pouvez choisir un nombre différent de résultats à afficher (jusqu'à 1 000) à l'aide de la liste déroulante Aperçu des 100 premiers résultats de requête.
7. Choisissez Télécharger les résultats pour télécharger vos résultats au format CSV. Vous pouvez télécharger jusqu'à 1 000 lignes de résultats.

Attacher un calcul à un espace de travail EMR Studio

Amazon EMR Studio exécute des commandes de bloc-notes à l'aide d'un noyau sur un cluster EMR. Avant de sélectionner un noyau, vous devez attacher l'espace de travail à un cluster qui utilise des instances Amazon EC2, à un cluster Amazon EMR sur EKS ou à une application EMR sans serveur. EMR Studio vous permet d'rattacher des Workspaces à des clusters nouveaux ou existants, et vous donne la possibilité de changer de cluster sans fermer le Workspace.

Cette section aborde les sujets suivants pour vous aider à travailler avec des clusters et à les mettre en service pour EMR Studio :

- Attacher un cluster Amazon EC2 à un Workspace EMR Studio

- Rattacher un cluster Amazon EMR à un Workspace EMR Studio
- Attacher une application Amazon EMR sans serveur à un espace de travail EMR Studio
- Création et rattachement d'un nouveau cluster EMR à un Workspace EMR Studio
- Détacher un calcul d'un espace de travail EMR Studio

Attacher un cluster Amazon EC2 à un Workspace EMR Studio

Vous pouvez rattacher un cluster EMR exécuté sur Amazon EC2 à un Workspace lorsque vous créez le Workspace, ou rattacher un cluster à un Workspace existant. Si vous souhaitez créer un nouveau cluster, consultez Création et rattachement d'un nouveau cluster EMR à un Workspace EMR Studio.

> **Note**
>
> Un Workspace d'un Studio où la propagation d'identité approuvée IAM Identity Center est activée ne peut être attaché qu'à un cluster EMR dont la configuration de sécurité inclut Identity Center activé.

On create

Connectez-vous à un cluster de calcul Amazon EMR lorsque vous créez un Workspace

1. Dans la boîte de dialogue Créer un Workspace, assurez-vous que vous avez déjà sélectionné un sous-réseau pour le nouvel Workspace. Développez la section Configuration avancée.
2. Choisissez Attacher le Workspace à un cluster EMR.
3. Dans la liste déroulante Clusters EMR, sélectionnez un cluster EMR existant à rattacher au Workspace.

Après avoir attaché un cluster, terminez la création du Workspace. Lorsque vous ouvrez le nouvel Workspace pour la première fois et que vous choisissez le panneau Clusters EMR, le cluster sélectionné devrait être rattaché.

On launch

Connectez-vous à un cluster de calcul Amazon EMR lorsque vous lancer le Workspace

1. Accédez à la liste des Workspaces et sélectionnez la ligne correspondant au Workspace que vous souhaitez lancer. Sélectionnez ensuite Lancer le Workspace **>** Lancer avec options.
2. Choisissez un cluster EMR à rattacher à votre Workspace.

Après avoir rattaché un cluster, terminez la création du Workspace. Lorsque vous ouvrez le nouvel Workspace pour la première fois et que vous choisissez le panneau Clusters EMR, le cluster sélectionné devrait être rattaché.

In JupyterLab

Associer un espace de travail à un cluster de calcul Amazon EMR dans JupyterLab

1. Sélectionnez votre Workspace, puis sélectionnez Lancer le Workspace **>** Lancement rapide.
2. À l'intérieur JupyterLab, ouvrez l'onglet Cluster dans la barre latérale gauche.
3. Sélectionnez le menu déroulant EMR sur un cluster EC2 ou sélectionnez un cluster Amazon EMR sur EKS.
4. Choisissez Attacher pour rattacher le cluster à votre Workspace.

Après avoir rattaché le cluster, terminez la création du Workspace. Lorsque vous ouvrez le nouvel Workspace pour la première fois et que vous choisissez le panneau Clusters EMR, le cluster sélectionné devrait être rattaché.

In the Workspace UI

Rattacher un Workspace à un cluster de calcul Amazon EMR depuis l'interface utilisateur du Workspace

1. dans le Workspace que vous souhaitez rattacher à un cluster, cliquez sur l'icône clusters EMR dans la barre latérale gauche pour ouvrir le panneau Cluster.
2. Sous Type de cluster, développez la liste déroulante et sélectionnez Cluster EMR sur EC2.
3. Dans la liste déroulante, choisissez un cluster. Il se peut que vous deviez d'abord détacher un cluster existant pour activer la liste déroulante de sélection des clusters.
4. Choisissez Attacher. Lorsque le cluster est rattaché, vous devriez voir apparaître un message de réussite.

Rattacher un cluster Amazon EMR à un Workspace EMR Studio

Outre l'utilisation de clusters Amazon EMR exécutés sur Amazon EC2, vous pouvez rattacher un Workspace à un cluster Amazon EMR sur EKS pour exécuter du code de bloc-notes. Pour plus d'informations sur Amazon EMR, veuillez consulter Qu'est-ce qu'Amazon EMR sur EKS ?.

Avant de pouvoir connecter un Workspace à un cluster Amazon EMR sur EKS, l'administrateur de votre studio doit vous accorder des autorisations d'accès.

> **Note**
>
> Vous ne pouvez pas lancer un cluster Amazon EMR sur EKS dans un Studio EMR qui utilise la propagation d'identité approuvée IAM Identity Center.

On create

Rattacher un cluster de calcul Amazon EMR lorsque vous créez un Workspace

1. Dans la boîte de dialogue Créer un Workspace, développez la section Configuration avancée.
2. Choisissez Attacher le Workspace à un cluster Amazon EMR sur EKS.
3. Sous Cluster Amazon EMR sur EKS, choisissez un cluster dans la liste déroulante.
4. Sous Sélectionner un point de terminaison, choisissez un point de terminaison géré à rattacher au Workspace. Un point de terminaison géré est une passerelle qui permet à EMR Studio de communiquer avec le cluster de votre choix.
5. Choisissez Créer un Workspace pour terminer le processus de création du Workspace et rattacher le cluster sélectionné.

Après avoir rattaché un cluster, vous pouvez terminer de créer le Workspace. Lorsque vous ouvrez le nouvel Workspace pour la première fois et que vous choisissez le panneau Clusters EMR, le cluster sélectionné devrait être rattaché.

In the Workspace UI

Rattacher un cluster Amazon EMR sur EKS depuis l'interface utilisateur du Workspace

1. dans le Workspace que vous souhaitez rattacher à un cluster, cliquez sur l'icône clusters EMR dans la barre latérale gauche pour ouvrir le panneau Cluster.
2. Développez la liste déroulante Type de cluster et choisissez Clusters EMR sur EKS.

3. Sous Cluster EMR sur EKS, choisissez un cluster dans la liste déroulante.
4. Sous Point de terminaison, choisissez un point de terminaison géré à rattacher au Workspace. Un point de terminaison géré est une passerelle qui permet à EMR Studio de communiquer avec le cluster de votre choix.
5. Choisissez Attacher. Lorsque le cluster est rattaché, vous devriez voir apparaître un message de réussite.

Attacher une application Amazon EMR sans serveur à un espace de travail EMR Studio

Vous pouvez attacher un espace de travail à une application EMR sans serveur pour exécuter des charges de travail interactives. Pour plus d'informations, consultez la rubrique relative à l'utilisation des blocs-notes pour exécuter des charges de travail interactives avec EMR sans serveur via EMR Studio.

> **Note**
>
> Vous ne pouvez pas attacher une application EMR sans serveur à un Studio EMR qui utilise la propagation d'identité approuvée IAM Identity Center.

Example Associer un espace de travail à une application EMR sans serveur dans JupyterLab

Avant de pouvoir connecter un espace de travail à une application EMR sans serveur, l'administrateur de votre compte doit vous accorder les autorisations d'accès décrites dans la rubrique Required permissions for interactive workloads.

1. Accédez à EMR Studio pour sélectionner votre espace de travail, puis sélectionnez Lancer une instance WorkSpace > Lancement rapide.
2. À l'intérieur JupyterLab, ouvrez l'onglet Cluster dans la barre latérale gauche.
3. Sélectionnez EMR sans serveur comme option de calcul, puis sélectionnez une application EMR sans serveur et un rôle d'exécution.
4. Pour attacher le cluster à votre espace de travail, choisissez Attacher.

Maintenant, lorsque vous ouvrez cet espace de travail, vous devriez voir l'application sélectionnée attachée.

Création et rattachement d'un nouveau cluster EMR à un Workspace EMR Studio

Les utilisateurs avancés d'EMR Studio peuvent configurer de nouveaux clusters EMR exécutés sur Amazon EC2 à utiliser avec un Workspace. Toutes les applications Big Data requises pour EMR Studio sont installées par défaut sur le nouveau cluster.

Pour créer des clusters, l'administrateur de votre studio doit d'abord vous donner l'autorisation à l'aide d'une politique de session. Pour plus d'informations, consultez Créer des politiques d'autorisation pour les utilisateurs d'EMR Studio.

Vous pouvez créer un nouveau cluster dans la boîte de dialogue Créer un Workspace ou depuis le panneau Cluster de l'interface utilisateur du Workspace. Dans les deux cas, vous avez le choix entre deux options de création de clusters :

1. Création d'un cluster EMR : créez un cluster EMR en choisissant le type et le nombre d'instances Amazon EC2.
2. Utiliser un modèle de cluster : mettez en service un cluster en sélectionnant un modèle de cluster prédéfini. Cette option apparaît si vous êtes autorisé à utiliser des modèles de cluster.

> **Note**
>
> Si vous avez activé la propagation d'identité approuvée avec IAM Identity Center pour votre Studio, vous devez utiliser un modèle pour créer un cluster.

Créer un cluster EMR en fournissant une configuration de cluster

1. Choisissez un point de départ.

Pour…	Faites ceci...
Créez le cluster lorsque vous créez un Workspace à l'aide de la boîte de dialogue Créer un Workspace.	Développez la section Configuration avancée dans la boîte de dialogue Créer un Workspace, puis sélectionnez Créer un cluster EMR.
Créez le cluster à partir du panneau du cluster EMR de l'interface utilisateur de Workspace après avoir créé un Workspace.	Choisissez l'onglet Clusters EMR dans la barre latérale gauche d'un Workspace ouvert,

Pour…	Faites ceci...
	développez la section Configuration avancée et choisissez Créer un cluster.

2. Entrez un nom de cluster. Le fait de nommer le cluster vous permet de le retrouver ultérieurement dans la liste des clusters EMR Studio.
3. Pour la version d'Amazon EMR, choisissez une version d'Amazon EMR pour le cluster.
4. Pour Instance, sélectionnez le type et le nombre d'instances Amazon EC2 pour le cluster. Pour plus d'informations sur le choix du type d'instance, consultez Configuration des instances Amazon EC2. Une instance est utilisée pour le nœud primaire.
5. Sélectionnez un sous-réseau dans lequel EMR Studio peut lancer le nouveau cluster. Chaque option de sous-réseau est préapprouvée par l'administrateur de votre studio, et votre Workspace doit être en mesure de se connecter à un cluster dans n'importe quel sous-réseau répertorié.
6. Choisissez un URI S3 pour le stockage des journaux.
7. Choisissez Créer un cluster EMR pour mettre en service le cluster. Si vous utilisez la boîte de dialogue Créer un Workspace, choisissez Créer un Workspace pour créer le Workspace et mettre en service le cluster. Une fois qu'EMR Studio a mis en service le nouveau cluster, il le rattache au Workspace.

Créer un cluster en utilisant un modèle de cluster

1. Choisissez un point de départ.

Pour…	Faites ceci...
Créez le cluster lorsque vous créez un Workspace à l'aide de la boîte de dialogue Créer un Workspace.	Développez la section Configuration avancée dans la boîte de dialogue Créer un Workspace, puis sélectionnez Utiliser un modèle de cluster.
Créez le cluster à partir du panneau du cluster EMR de l'interface utilisateur de Workspace.	Choisissez l'onglet Clusters EMR dans la barre latérale gauche d'un Workspace ouvert, développez la section Configuration avancée et choisissez Modèle de cluster.

2. Sélectionnez un modèle de cluster dans la liste déroulante. Chaque modèle de cluster disponible inclut une brève description pour vous aider à effectuer une sélection.
3. Le modèle de cluster que vous choisissez peut comporter des paramètres supplémentaires tels que la version de version d'Amazon EMR ou le nom du cluster. Vous pouvez choisir ou insérer des valeurs, ou utiliser les valeurs par défaut sélectionnées par votre administrateur.
4. Sélectionnez un sous-réseau dans lequel EMR Studio peut lancer le nouveau cluster. Chaque option de sous-réseau est préapprouvée par l'administrateur de votre studio, et votre Workspace doit être en mesure de se connecter à un cluster dans n'importe quel sous-réseau.
5. Choisissez Utiliser un modèle de cluster pour mettre en service le cluster et le rattacher au Workspace. Il faudra quelques minutes à EMR Studio pour créer le cluster. Si vous utilisez la boîte de dialogue Créer un Workspace, choisissez Créer un Workspace pour créer le Workspace et mettre en service le cluster. Une fois qu'EMR Studio a mis en service le nouveau cluster, il le rattache à votre Workspace.

Détacher un calcul d'un espace de travail EMR Studio

Pour échanger le cluster rattaché à un Workspace, vous pouvez détacher un cluster de l'interface utilisateur du Workspace.

Détacher un cluster d'un Workspace

1. dans le Workspace que vous souhaitez détacher d'un cluster, cliquez sur l'icône clusters EMR dans la barre latérale gauche pour ouvrir le panneau Cluster.
2. Sous Sélectionner un cluster, choisissez Détacher et attendez qu'EMR Studio détache le cluster. Lorsque le cluster est détaché, un message de réussite s'affiche.

Pour détacher une application EMR sans serveur d'un espace de travail EMR Studio

Pour échanger le calcul attaché à un espace de travail, vous pouvez détacher l'application de l'interface utilisateur de l'espace de travail.

1. Dans l'espace de travail que vous souhaitez détacher d'un cluster, cliquez sur l'icône de calcul Amazon EMR dans la barre latérale gauche pour ouvrir le panneau Calcul.
2. Sous Sélectionner un calcul, choisissez Détacher et attendez qu'EMR Studio détache l'application. Lorsque l'application est détachée, un message de réussite s'affiche.

Lier des référentiels Git à un Workspace EMR Studio

À propos des référentiels Git pour EMR Studio

Vous pouvez rattacher un maximum de trois référentiels Git à un Workspace EMR Studio. Par défaut, chaque espace de travail vous permet de choisir parmi une liste de référentiels Git associés au même AWS compte que le Studio. Vous pouvez également créer un nouveau référentiel Git en tant que ressource pour un Workspace.

Vous pouvez exécuter des commandes Git comme suit à l'aide d'une commande de terminal lorsque vous êtes connecté au nœud primaire d'un cluster.

```
!git pull origin <branch-name>
```

Vous pouvez également utiliser l'extension jupyterlab-git. Ouvrez-la depuis la barre latérale gauche en choisissant l'icône Git. Pour plus d'informations sur l'extension jupyterlab-git pour, consultez jupyterlab-git. JupyterLab

Prérequis

- Pour rattacher un référentiel Git à un Workspace, le studio doit être configuré pour autoriser la liaison entre les référentiels Git. L'administrateur de votre studio doit prendre les mesures nécessaires pour Établissez l'accès et les autorisations pour les référentiels Git.
- Si vous utilisez un CodeCommit dépôt, vous devez utiliser les informations d'identification Git et HTTPS. Les clés SSH et le protocole HTTPS avec l'assistant AWS Command Line Interface d'identification ne sont pas pris en charge. CodeCommit ne prend pas non plus en charge les jetons d'accès personnels (PAT). Pour plus d'informations, consultez les sections Utilisation d'IAM avec CodeCommit dans le guide de l'utilisateur IAM et Configuration pour les utilisateurs HTTPS à l'aide des informations d'identification Git dans le guide de l'AWS CodeCommit utilisateur.

Instructions

Pour lier un référentiel Git rattaché à un Workspace

1. Ouvrez le Workspace que vous souhaitez lier à un référentiel depuis la liste Workspaces du studio.
2. Dans la barre latérale gauche, choisissez l'icône Référentiel Git Amazon EMR pour ouvrir le panneau d'outils du référentiel Git.

3. Sous Référentiels Git, développez la liste déroulante et sélectionnez un maximum de trois référentiels à rattacher au Workspace. EMR Studio enregistre votre sélection et commence à lier chaque référentiel.

Le processus de liaison peut prendre un certain temps. Vous pouvez voir l'état de chaque référentiel que vous avez sélectionné dans le panneau d'outils Référentiel Git. Une fois qu'EMR Studio a lié un référentiel à un Workspace, les fichiers appartenant à ce référentiel devraient apparaître dans le panneau Navigateur de fichiers.

Ajouter un nouveau référentiel Git à un Workspace en tant que ressource

1. Ouvrez le Workspace que vous souhaitez lier à un référentiel depuis la liste Workspaces de votre studio.
2. Dans la barre latérale gauche, choisissez l'icône Référentiel Git Amazon EMR pour ouvrir le panneau d'outils du référentiel Git.
3. Choisissez Ajouter un nouveau référentiel Git.
4. Pour Nom du référentiel, entrez un nom descriptif à utiliser pour le référentiel dans EMR Studio. Les noms ne peuvent contenir que des caractères alphanumériques, des traits d'union ou des traits de soulignement.
5. Pour URL du référentiel Git, entrez l'URL du référentiel. Lorsque vous utilisez un CodeCommit référentiel, il s'agit de l'URL qui est copiée lorsque vous choisissez Cloner l'URL puis Cloner HTTPS. Par exemple, `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/[MyCodeCommitRepoName]`.
6. Pour Branche, entrez le nom d'une branche existante que vous souhaitez récupérer.
7. Pour les informations d'identification Git, choisissez une option selon les instructions suivantes. EMR Studio accède à vos informations d'identification Git à l'aide des secrets stockés dans Secrets Manager.

> **Note**
>
> Si vous utilisez un GitHub référentiel, nous vous recommandons d'utiliser un jeton d'accès personnel (PAT) pour vous authentifier. À compter du 13 août 2021, une authentification basée sur des jetons GitHub sera requise et les mots de passe ne seront plus acceptés lors de l'authentification des opérations Git. Pour plus d'informations,

consultez l'article sur les exigences d'authentification par jeton pour les opérations Git dans The GitHub Blog.

Option	Description
Création d'un secret	Choisissez cette option pour associer les informations d'identification Git existantes à un nouveau secret qui sera créé AWS Secrets Manager pour vous. 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, entrez le nom secret à utiliser dans Secrets Manager, puis indiquez le nom d'utilisateur et le mot de passe à rattacher au secret. – OU – Si vous utilisez un jeton d'accès personnel pour accéder au référentiel, sélectionnez Jeton d'accès personnel (PAT), saisissez le nom du secret à utiliser dans Secrets Manager, puis saisissez votre jeton d'accès personnel. Pour plus d'informations, consultez Création d'un jeton d'accès personnel pour la ligne de commande GitHub et de jetons d'accès personnels pour Bitbucket. CodeCommit les référentiels ne prennent pas en charge cette option.

Option	Description
Utilisation d'un référentiel public sans informations d'identification	Choisissez cette option pour accéder à un référentiel public.
Utiliser un AWS secret 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 rattaché à un nom d'utilisateur et un mot de passe Git, le secret doit être au format `{"gitUsername": " MyUserName ", "gitPassword": " MyPassword "}`.

8. Choisissez Ajouter un référentiel pour créer le nouveau référentiel. Une fois qu'EMR Studio a créé le nouveau référentiel, vous verrez un message de confirmation. Le nouveau référentiel apparaît dans la liste déroulante des référentiels Git.
9. Pour lier le nouveau référentiel à votre Workspace, sélectionnez-le dans la liste déroulante située sous Référentiels Git.

Le processus de liaison peut prendre un certain temps. Une fois qu'EMR Studio a lié le nouveau référentiel au Workspace, un nouveau dossier portant le même nom que votre référentiel devrait apparaître dans le panneau de Navigateur de fichiers.

Pour ouvrir un autre référentiel lié, accédez au dossier correspondant dans le navigateur de fichiers.

Utilisation de l’éditeur SQL Amazon Athena dans EMR Studio

Présentation

Vous pouvez utiliser Amazon EMR Studio pour développer et exécuter des requêtes interactives sur Amazon Athena. De cette façon, vous pouvez effectuer des analytiques SQL sur Athena depuis la même interface EMR Studio que celle que vous utilisez pour exécuter vos charges de travail Spark, Scala et autres. Grâce à cette intégration, vous pouvez utiliser la saisie automatique pour développer rapidement des requêtes, parcourir les données de votre catalogue de données AWS Glue, créer des requêtes enregistrées, consulter l'historique de vos requêtes, etc.

Pour plus d'informations sur l'utilisation d'Amazon Athena, voir la rubrique Utilisation d'Athena SQL du Guide de l'utilisateur Amazon Athena.

Utilisation de l'éditeur SQL Athena dans EMR Studio

Procédez comme suit pour développer et exécuter des requêtes interactives sur Amazon Athena depuis votre Studio EMR :

1. Ajoutez les autorisations requises au rôle d'utilisateur pour les utilisateurs qui accèdent aux Workspaces de ce Studio. Les autorisations sont répertoriées dans le tableau dans la colonne Accès à l'éditeur SQL Amazon Athena depuis votre Studio EMR Studio du tableau Autorisations AWS Identity and Access Management pour les utilisateurs d'EMR Studio. Vous pouvez également choisir de copier le contenu de la politique avancée sur la page Exemple de politiques utilisateur pour accorder aux utilisateurs des autorisations complètes sur les fonctionnalités d'EMR Studio, y compris celle-ci.
2. Configurez et créez un Studio EMR.
3. Accédez à votre Studio et sélectionnez Éditeur de requêtes dans la barre latérale.

Vous devriez désormais voir l'interface utilisateur habituelle de l'éditeur Athena. Pour plus d'informations sur la prise en main et l'utilisation d'Athena SQL pour exécuter des requêtes interactives, consultez les rubriques Mise en route et Utilisation d'Athena SQL du Guide de l'utilisateur Amazon Athena.

> **Note**
>
> Si vous avez activé la propagation d'identité approuvée via IAM Identity Center pour votre Studio EMR, vous devez utiliser les groupes de travail Athena pour contrôler l'accès aux requêtes, et le groupe de travail que vous utilisez doit également utiliser la propagation d'identité approuvée. Pour savoir comment configurer Identity Center et activer la propagation d'identité approuvée pour votre groupe de travail, voir la rubrique Utilisation des groupes de travail Athena compatibles avec IAM Identity Center du Guide de l'utilisateur Amazon Athena.

Considérations relatives à l'utilisation de l'éditeur SQL Athena dans EMR Studio

- L'intégration à Athena est disponible dans toutes les régions commerciales où EMR Studio et Athena sont disponibles.

- Les fonctionnalités Athena suivantes ne sont pas disponibles dans EMR Studio :
  - Fonctionnalités administrateur telles que la création ou la mise à jour de groupes de travail, de sources de données ou de réserves de capacité Athena
  - Athena pour Spark ou blocs-notes Spark
  - DataZone Intégration avec Amazon
  - Optimiseur basé sur les coûts (CBO)
  - Step Functions

CodeWhisperer Intégration d'Amazon à EMR Studio Workspaces

Présentation

Vous pouvez utiliser Amazon CodeWhisperer avec Amazon EMR Studio pour obtenir des recommandations en temps réel lorsque vous écrivez du code. JupyterLab CodeWhispererpeut compléter vos commentaires, terminer des lignes de code individuelles, faire line-by-line des recommandations et générer des fonctions entièrement formées.

> **Note**
>
> Lorsque vous utilisez Amazon EMR Studio, il est possible que vous AWS stockiez des données relatives à votre utilisation et à votre contenu à des fins d'amélioration du service. Pour plus d'informations et des instructions pour désactiver le partage de données, consultez la section Partage de vos données avec AWS dans le guide de CodeWhisperer l'utilisateur Amazon.

Considérations relatives à l'utilisation CodeWhisperer avec Workspaces

- CodeWhisperer l'intégration est disponible Régions AWS là où EMR Studio est disponible, comme indiqué dans les considérations relatives à EMR Studio.
- Amazon EMR Studio utilise automatiquement le point de CodeWhisperer terminaison situé dans l'est des États-Unis (Virginie du Nord) (us-east-1) pour les recommandations, quelle que soit la région dans laquelle se trouve votre studio.
- CodeWhisperer prend uniquement en charge le langage Python pour le codage de scripts ETL pour les tâches Spark dans EMR Studio.

- Une option de télémétrie côté client quantifie votre utilisation de. CodeWhisperer Cette fonctionnalité n’est pas prise en charge avec EMR Studio.

Autorisations requises pour CodeWhisperer

Pour l'utiliser CodeWhisperer, vous devez associer la politique suivante à votre rôle d'utilisateur IAM pour Amazon EMR Studio :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
          "Sid": "CodeWhispererPermissions",
          "Effect": "Allow",
          "Action": [ "codewhisperer:GenerateRecommendations" ],
          "Resource": "*"
        }
     ]
}
```

Utilisation CodeWhisperer avec les espaces de travail

Pour afficher le journal de CodeWhisperer référence JupyterLab, ouvrez le CodeWhispererpanneau en bas de la JupyterLab fenêtre et choisissez Ouvrir le journal de référence du code.

La liste suivante contient des raccourcis que vous pouvez utiliser pour interagir avec les CodeWhisperer suggestions :

- Recommandations de pause — Utilisez les suggestions automatiques de pause dans les CodeWhisperer paramètres.
- Accepter une recommandation : appuyez sur la touche Tab de votre clavier.
- Rejeter une recommandation : appuyez sur la touche Echap de votre clavier.
- Parcourir les recommandations : utilisez les flèches haut et bas de votre clavier.
- Invocation manuelle : appuyez sur les touches Alt et C de votre clavier. Si vous utilisez un Mac, appuyez sur les touches Cmd et C.

Vous pouvez également l'utiliser CodeWhisperer pour modifier des paramètres tels que le niveau de journalisation et obtenir des suggestions de références de code. Pour plus d'informations, consultez

la section Configuration CodeWhisperer JupyterLab et fonctionnalités du guide de CodeWhisperer l'utilisateur Amazon.

Déboguer des applications et des tâches avec EMR Studio

Avec Amazon EMR Studio, vous pouvez lancer des interfaces d'applications de données pour analyser les applications et les exécutions de tâches dans le navigateur.

Vous pouvez également lancer les interfaces utilisateur persistantes hors cluster pour Amazon EMR exécuté sur des clusters EC2 depuis la console Amazon EMR. Pour plus d'informations, consultez Afficher les interfaces utilisateur d'application persistante.

> **Note**
>
> Selon les paramètres de votre navigateur, il se peut que vous deviez activer les fenêtres contextuelles pour que l'interface utilisateur d'une application s'ouvre.

Pour plus d'informations sur la configuration et l'utilisation des interfaces d'application, consultez Serveur de chronologie YARN, Surveillance et instrumentation, ou Aperçu de l'interface utilisateur Tez.

Déboguer Amazon EMR en cours d'exécution sur des tâches Amazon EC2

Workspace UI

Lancer une interface utilisateur intégrée au cluster à partir d'un fichier de bloc-notes

Lorsque vous utilisez les versions 5.33.0 et ultérieures d'Amazon EMR, vous pouvez lancer l'interface utilisateur Web de Spark (l'interface utilisateur Spark ou le serveur d'historique Spark) depuis un bloc-notes de votre Workspace.

Les interfaces utilisateur intégrées au cluster fonctionnent avec PySpark les noyaux Spark ou SparkR. La taille de fichier maximale consultable pour les journaux d'événements ou les journaux de conteneurs Spark est de 10 Mo. Si vos fichiers journaux dépassent 10 Mo, nous vous recommandons d'utiliser le serveur d'historique Spark persistant plutôt que l'interface utilisateur Spark intégrée au cluster pour déboguer les tâches.

> **Important**
>
> Pour qu'EMR Studio puisse lancer des interfaces utilisateur d'applications sur un cluster à partir d'un Workspace, le cluster doit être en mesure de communiquer avec Amazon API Gateway. Vous devez configurer le cluster EMR pour autoriser le trafic réseau sortant vers Amazon API Gateway et vous assurer qu'Amazon API Gateway est accessible depuis le cluster.
> L'interface utilisateur de Spark accède aux journaux des conteneurs en résolvant les noms d'hôte. Si vous utilisez un nom de domaine personnalisé, vous devez vous assurer que les noms d'hôte de vos nœuds de cluster peuvent être résolus par Amazon DNS ou par le serveur DNS que vous spécifiez. Pour ce faire, définissez les options du protocole de configuration d'hôte dynamique (DHCP) pour le cloud privé virtuel (VPC) Amazon rattaché à votre cluster. Pour plus d'informations sur les options DHCP, consultez Ensembles d'options DHCP dans le Guide de l'utilisateur du cloud privé virtuel (VPC) Amazon.

1. Dans votre EMR Studio, ouvrez le Workspace que vous souhaitez utiliser et assurez-vous qu'il est rattaché à un cluster Amazon EMR exécuté sur EC2. Pour obtenir des instructions, veuillez consulter Attacher un calcul à un espace de travail EMR Studio.
2. Ouvrez un fichier bloc-notes et utilisez PySpark le noyau Spark ou SparkR. Pour sélectionner un noyau, choisissez le nom du noyau dans le coin supérieur droit de la barre d'outils du bloc-notes pour ouvrir la boîte de dialogue Sélectionner le noyau. Si aucun noyau n'a été sélectionné, le nom affiché est No Kernel!.
3. Exécutez le code de votre bloc-notes. Ce qui suit apparaît sous forme de sortie dans le bloc-notes lorsque vous démarrez le contexte Spark. L'affichage de l'information peut prendre quelques secondes. Si vous avez démarré le contexte Spark, vous pouvez exécuter la commande `%%info` pour accéder à un lien vers l'interface utilisateur de Spark à tout moment.

> **Note**
>
> Si les liens de l'interface utilisateur Spark ne fonctionnent pas ou n'apparaissent pas au bout de quelques secondes, créez une nouvelle cellule de bloc-notes et exécutez la commande `%%info` pour régénérer les liens.

4. Pour lancer l'interface utilisateur Spark, choisissez Lien sous Interface utilisateur Spark. Si votre application Spark est en cours d'exécution, l'interface utilisateur de Spark s'ouvre dans un nouvel onglet. Si l'application est terminée, le serveur d'historique Spark s'ouvre à la place.

   Après avoir lancé l'interface utilisateur Spark, vous pouvez modifier l'URL dans le navigateur pour ouvrir le serveur YARN ResourceManager ou Yarn Timeline. Ajoutez ensuite l'un des chemins suivants après `amazonaws.com`.

Interface utilisateur Web	Chemin	Exemple d'URL modifiée
FIL ResourceManager	/rm	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*rm*
Serveur de chronologie YARN	/yts	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*yts*
Serveur d'historique Spark	/shs	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com/*shs*

Studio UI

Lancez le serveur de chronologie YARN persistant, le serveur d'historique Spark ou l'interface utilisateur Tez depuis l'interface utilisateur d'EMR Studio

1. Dans votre EMR Studio, sélectionnez Amazon EMR sur EC2 sur le côté gauche de la page pour ouvrir la liste des clusters Amazon EMR sur EC2.
2. Filtrez la liste des clusters par nom, état ou ID en saisissant des valeurs dans le champ de recherche. Vous pouvez également effectuer une recherche par plage horaire de création.
3. Sélectionnez un cluster, puis choisissez Lancer l'interface utilisateur de l'application pour sélectionner une interface utilisateur d'application. L'interface utilisateur de l'application s'ouvre dans un nouvel onglet de navigateur ; le chargement peut prendre un certain temps.

Déboguer EMR Studio exécuté sur EMR sans serveur

À l'instar d'Amazon EMR exécuté sur Amazon EC2, vous pouvez utiliser l'interface utilisateur de l'espace utilisateur pour analyser vos applications EMR sans serveur. Depuis l'interface utilisateur de l'espace de travail, lorsque vous utilisez les versions 6.14.0 et ultérieures d’Amazon EMR, vous pouvez lancer l'interface utilisateur web de Spark (l'interface utilisateur Spark ou le serveur d'historique Spark) depuis un bloc-notes de votre espace de travail. Pour vous faciliter la tâche, nous fournissons également un lien vers le journal de pilote pour accéder rapidement aux journaux de pilote Spark.

Déboguer Amazon EMR sur une tâche EKS exécutée avec le serveur d'historique Spark

Lorsque vous soumettez une tâche exécutée à un cluster Amazon EMR sur EKS, vous pouvez accéder aux journaux de cette tâche exécutée à l'aide du serveur d'historique Spark. Le serveur d'historique Spark fournit des outils pour surveiller les applications Spark, tels qu'une liste des étapes et des tâches du planificateur, un récapitulatif des tailles RDD et de l'utilisation de la mémoire, ainsi que des informations environnementales. Vous pouvez lancer le serveur d'historique Spark pour Amazon EMR lorsque les tâches EKS sont exécutées de la manière suivante :

- Lorsque vous soumettez une tâche exécutée à l'aide d'EMR Studio avec un point de terminaison géré par Amazon EMR on EKS, vous pouvez lancer le serveur d'historique Spark à partir d'un fichier bloc-notes dans votre Workspace.

- Lorsque vous soumettez une tâche exécutée à l'aide du AWS SDK AWS CLI ou du SDK pour Amazon EMR sur EKS, vous pouvez lancer le serveur Spark History depuis l'interface utilisateur d'EMR Studio.

Pour plus d'informations sur l'utilisation du serveur d'historique Spark, consultez Surveillance et instrumentation dans la documentation d'Apache Spark. Pour plus d'informations sur les exécutions de tâches, consultez Concepts et composants dans le Guide de développement Amazon EMR sur EKS.

Lancer le serveur d'historique Spark à partir d'un fichier bloc-notes dans votre Workspace EMR Studio

1. Ouvrez un Workspace connecté à un cluster Amazon EMR sur EKS.
2. Sélectionnez et ouvrez le fichier de votre bloc-notes dans le Workspace.
3. Choisissez Interface utilisateur Spark en haut du fichier de bloc-notes pour ouvrir le serveur d'historique Spark persistant dans un nouvel onglet.

Lancer le serveur d'historique Spark depuis l'interface utilisateur d'EMR Studio

> **Note**
>
> La liste des tâches de l'interface utilisateur d'EMR Studio affiche uniquement les exécutions de tâches que vous soumettez à l'aide du AWS SDK AWS CLI ou du SDK pour Amazon EMR sur EKS.

1. Dans votre EMR Studio, sélectionnez Amazon EMR sur EKS sur le côté gauche de la page.
2. Recherchez le cluster virtuel Amazon EMR sur EKS que vous avez utilisé pour soumettre votre exécution de tâche. Vous pouvez filtrer la liste des clusters par état ou par ID en saisissant des valeurs dans le champ de recherche.
3. Sélectionnez le cluster pour ouvrir sa page de détails. La page détaillée affiche des informations sur le cluster, telles que l'ID, l'espace de noms et l'état. La page affiche également une liste de toutes les exécutions de tâches soumises à ce cluster.
4. Sur la page détaillée du cluster, sélectionnez une tâche à exécuter pour le déboguer.
5. Dans le coin supérieur droit de la liste des tâches, choisissez Lancer le serveur d'historique Spark pour ouvrir l'interface de l'application dans un nouvel onglet du navigateur.

Installation de noyaux et de bibliothèques dans un Workspace EMR Studio

Chaque Amazon EMR Studio est livré avec un ensemble de bibliothèques et de noyaux pré-installés.

Noyaux et bibliothèques sur les clusters exécutés sur Amazon EC2

Lorsque vous utilisez des clusters EMR exécutés sur Amazon EC2, vous pouvez également personnaliser l'environnement d'EMR Studio de la manière suivante :

- Installer les noyaux bloc-notes Jupyter et les bibliothèques Python sur un nœud primaire du cluster : lorsque vous installez des bibliothèques à l'aide de cette option, tous les Workspaces rattachés au même cluster partagent ces bibliothèques. Vous pouvez installer des noyaux ou des bibliothèques à partir d'une cellule de bloc-notes ou lorsque vous êtes connecté via SSH au nœud primaire d'un cluster.
- Utiliser des bibliothèques adaptées à un bloc-notes : lorsque les utilisateurs de Workspace installent et utilisent des bibliothèques depuis une cellule de bloc-notes, ces bibliothèques ne sont disponibles que pour ce bloc-notes. Cette option permet à différents blocs-notes utilisant le même cluster de fonctionner sans se soucier des conflits de versions de bibliothèque.

Les Workspaces EMR Studio ont la même architecture sous-jacente que les bloc-notes EMR. Vous pouvez installer et utiliser les noyaux bloc-notes Jupyter ainsi que les bibliothèques Python avec EMR Studio de la même manière que vous le feriez avec les bloc-notes EMR. Pour obtenir des instructions, veuillez consulter Installation et utilisation des noyaux et des bibliothèques.

Noyaux et bibliothèques sur les clusters Amazon EMR sur EKS

Les clusters Amazon EMR sur EKS incluent les noyaux et PySpark Python 3.7 avec un ensemble de bibliothèques préinstallées. Amazon EMR sur EKS ne prend pas en charge l'installation de bibliothèques ou de clusters supplémentaires.

Chaque cluster Amazon EMR on EKS est livré avec le Python et les PySpark bibliothèques suivants installés :

- Python – boto3, cffi, future, ggplot, jupyter, kubernetes, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn
- PySpark – ggplot, jupyter, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn

Noyaux et bibliothèques sur les applications EMR sans serveur

Chaque application EMR Serverless est livrée avec le Python et PySpark les bibliothèques suivants installés :

- Python – ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn
- PySpark – ggplot, matplotlib,numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn

Améliorer les noyaux avec des commandes magic

Présentation

EMR Studio et les blocs-notes EMR prennent en charge les commandes magic. Les commandes Magic, ou les magics, sont des améliorations apportées par le noyau IPython pour faciliter l'exécution et l'analyse des données. IPython est un environnement shell interactif qui est construit avec Python.

Amazon EMR prend également en charge Sparkmagic un package qui fournit des commandes magic spécifiques aux noyaux liés à Spark (PySparknoyaux SparkR et Scala) et qui utilise Livy sur le cluster pour soumettre des tâches Spark.

Tant que vous avez un noyau Python dans votre bloc-notes EMR, vous pouvez utiliser des commandes magic. De même, tout noyau lié à Spark prend en charge les commandes Sparkmagic.

Les commandes Magic, également appelées magics, se déclinent en deux variétés :

- magics des lignes : ces commandes magic sont désignées par un seul préfixe `%` et fonctionnent sur une seule ligne de code
- magic des cellules : ces commandes magic sont désignées par un double préfixe `%%` et fonctionnent sur plusieurs lignes de code

Pour connaître toutes les Répertorier les commandes magic et Sparkmagics disponibles, consultez magic.

Considérations et restrictions

- EMR sans serveur ne prend pas en charge `%%sh` pour exécuter `spark-submit`. Il ne prend pas en charge les magics de blocs-notes EMR.
- Amazon EMR sur les clusters EKS ne prend pas en charge les commandes Sparkmagic pour EMR Studio. Cela est dû au fait que les noyaux Spark que vous utilisez avec des points de terminaison

gérés sont intégrés à Kubernetes et ne sont pas pris en charge par Sparkmagic et Livy. Vous pouvez définir la configuration Spark directement dans l' SparkContext objet pour contourner le problème, comme le montre l'exemple suivant.

```
spark.conf.set("spark.driver.maxResultSize", '6g')
```

- Les magic commandes et actions suivantes sont interdites par AWS :
  - `%alias`
  - `%alias_magic`
  - `%automagic`
  - `%macro`
  - Modifier `proxy_user` avec `%configure`
  - Modifier `KERNEL_USERNAME` avec `%env` ou `%set_env`

Répertorier les commandes magic et Sparkmagic

Utilisez les commandes suivantes pour répertorier les commandes magic disponibles :

- `%lsmagic` répertorie toutes les fonctions magic actuellement disponibles ;
- `%%help` répertorie les fonctions magic liées à Spark actuellement disponibles fournies par le package Sparkmagic.

Utiliser **`%%configure`** pour configurer Spark

L'une des commandes Sparkmagic les plus utiles est la commande `%%configure`, qui configure les paramètres de création de session. Pour les paramètres `conf`, vous pouvez configurer n'importe quelle configuration Spark mentionnée dans la documentation de configuration d'Apache Spark.

Example Ajouter un fichier JAR externe à EMR Notebooks depuis le référentiel Maven ou Amazon S3

Vous pouvez utiliser l'approche suivante pour ajouter une dépendance à un fichier JAR externe à tout noyau lié à Spark pris en charge par Sparkmagic.

```
%%configure -f
{"conf": {
    "spark.jars.packages": "com.jsuereth:scala-arm_2.11:2.0,ml.combust.bundle:bundle-
ml_2.11:0.13.0,com.databricks:dbutils-api_2.11:0.0.3",
```

```
    "spark.jars": "s3://DOC-EXAMPLE-BUCKET/my-jar.jar"
    }
}
```

Example : Configurer Hudi

Vous pouvez utiliser l'éditeur de bloc-notes pour configurer votre bloc-notes EMR afin d'utiliser Hudi.

```
%%configure
{ "conf": {
     "spark.jars": "hdfs://apps/hudi/lib/hudi-spark-bundle.jar,hdfs:///apps/hudi/lib/
spark-spark-avro.jar",
     "spark.serializer": "org.apache.spark.serializer.KryoSerializer",
     "spark.sql.hive.convertMetastoreParquet":"false"
     }
}
```

Utiliser **`%%sh`** pour exécuter **`spark-submit`**

La magic `%%sh` exécute des commandes shell dans un sous-processus sur une instance de votre cluster rattaché. Généralement, vous utilisez l'un des noyaux liés à Spark pour exécuter des applications Spark sur votre cluster attaché. Toutefois, si vous souhaitez utiliser un noyau Python pour soumettre une application Spark, vous pouvez utiliser la magic suivante, en remplaçant le nom du compartiment par le nom de votre compartiment en minuscules.

```
%%sh
spark-submit --master yarn --deploy-mode cluster s3://DOC-EXAMPLE-BUCKET/test.py
```

Dans cet exemple, le cluster doit accéder à l'emplacement de `s3://DOC-EXAMPLE-BUCKET/test.py`, sinon la commande échouera.

Vous pouvez utiliser n'importe quelle commande Linux avec la magic `%%sh`. Si vous souhaitez exécuter des commandes Spark ou YARN, utilisez l'une des options suivantes pour créer un utilisateur Hadoop `emr-notebook` et accordez-lui les autorisations nécessaires pour exécuter les commandes :

- Vous pouvez créer un nouvel utilisateur de manière explicite en exécutant les commandes suivantes.

  ```
  hadoop fs -mkdir /user/emr-notebook
  ```

```
hadoop fs -chown emr-notebook /user/emr-notebook
```

- Vous pouvez activer une identité utilisateur dans Livy, qui créera automatiquement l'utilisateur. Pour plus d'informations, consultez Activation de l'emprunt d'identité pour contrôler l'activité des utilisateurs et des tâches Spark.

Utiliser `%%display` pour visualiser les dataframes Spark

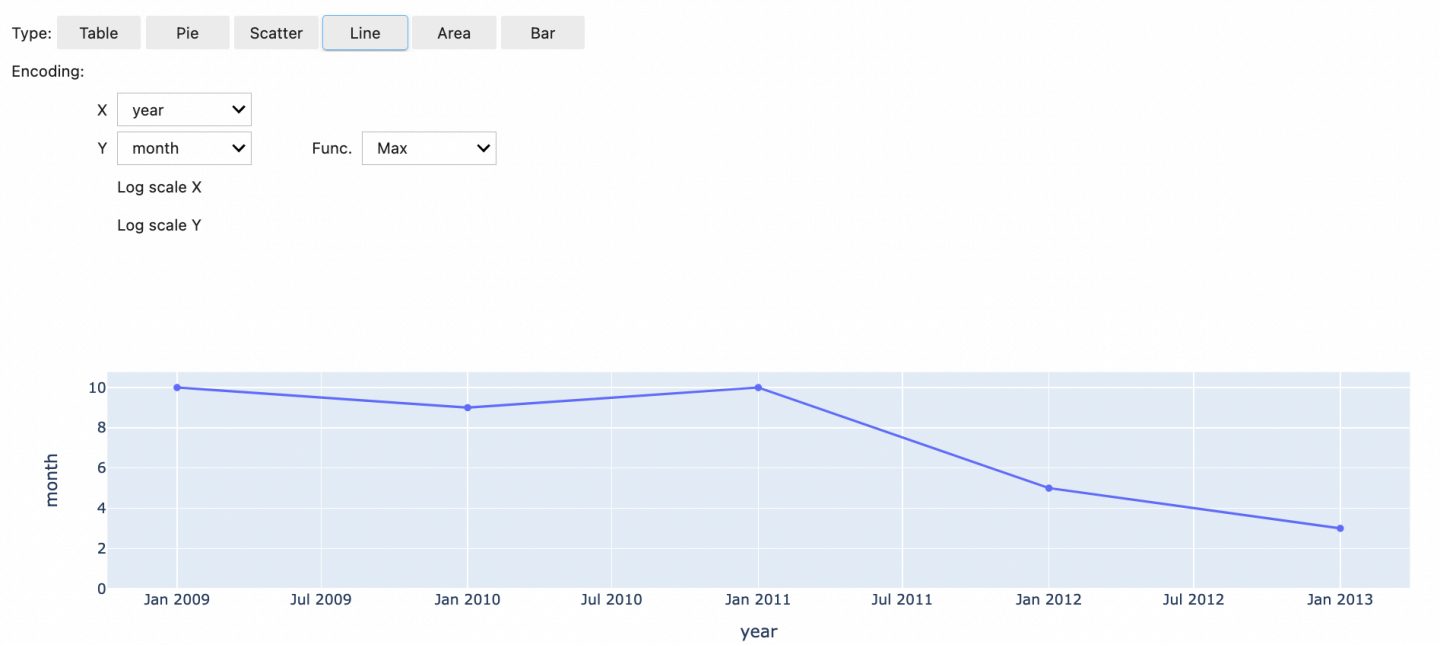
Vous pouvez utiliser la magic `%%display` pour visualiser une dataframe Spark. Pour utiliser cette magic, exécutez la commande suivante.

```
%%display df
```

Choisissez d'afficher les résultats sous forme de tableau, comme le montre l'image suivante.

year	month	total_passengers	total_trips
2012-01-01	3	26866837	16146923
2011-01-01	3	26091246	16066350
2013-01-01	3	26965079	15749228
2011-01-01	10	26287953	15707756
2009-01-01	10	26202049	15604551
2012-01-01	5	26278817	15567525
2011-01-01	5	25508952	15554868
2010-01-01	9	25533166	15540209
2010-01-01	5	26002858	15481351
2012-01-01	4	25900645	15477914

Vous pouvez également choisir de visualiser vos données à l'aide de cinq types de graphiques. Vous avez le choix entre des diagrammes circulaires, des diagrammes de dispersion, des diagrammes linéaires, des diagrammes de surface et des diagrammes à barres.



Utiliser les magics des blocs-notes EMR

Amazon EMR fournit les magics des blocs-notes EMR suivantes ; vous pouvez les utiliser avec les noyaux basés sur Python3 et Spark :

- `%mount_workspace_dir` : monte votre répertoire Workspace sur votre cluster afin que vous puissiez importer et exécuter du code à partir d'autres fichiers de votre Workspace

> **Note**
>
> Avec `%mount_workspace_dir`, seul le noyau Python 3 peut accéder à vos systèmes de fichiers locaux. Les exécuteurs Spark n'auront pas accès au répertoire monté avec ce noyau.

- `%umount_workspace_dir` : démonte votre répertoire Workspace de votre cluster
- `%generate_s3_download_url` : génère un lien de téléchargement temporaire dans la sortie de votre bloc-notes pour un objet Amazon S3

Prérequis

Avant d'installer les magics des blocs-notes EMR, effectuez les tâches suivantes :

- Assurez-vous que votre Rôle de service pour les instances EC2 de cluster (profil d'instance EC2) dispose d'un accès en lecture pour Amazon S3. Le `EMR_EC2_DefaultRole` avec la politique gérée `AmazonElasticMapReduceforEC2Role` répond à cette exigence. Si vous utilisez une politique ou un rôle personnalisés, assurez-vous qu'ils disposent des autorisations S3 nécessaires.

  > **Note**
  >
  > Les magic des blocs-notes EMR s'exécutent sur un cluster en tant qu'utilisateur du bloc-notes et utilisent le profil d'instance EC2 pour interagir avec Amazon S3. Lorsque vous montez un répertoire Workspace sur un cluster EMR, tous les Workspaces et blocs-notes EMR autorisés à se connecter à ce cluster peuvent accéder au répertoire monté. Les répertoires sont montés en lecture seule par défaut. Bien que `s3fs-fuse` et `goofys` autorisent les montages en lecture-écriture, nous vous recommandons vivement de ne pas modifier les paramètres de montage pour monter des répertoires en mode lecture-écriture. Si vous autorisez l'accès en écriture, toutes les modifications apportées au répertoire sont enregistrées dans le compartiment S3. Pour éviter la suppression ou le remplacement accidentels, vous pouvez activer la gestion des versions pour votre compartiment S3. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

- Exécutez l'un des scripts suivants sur votre cluster pour installer les dépendances pour les magics des blocs-notes EMR. Pour exécuter un script, vous pouvez Utilisation d'actions d'amorçage personnalisées ou suivre les instructions de la section Exécuter des commandes et des scripts sur un cluster Amazon EMR lorsque vous avez déjà un cluster en cours d'exécution.

  Vous pouvez choisir la dépendance à installer. s3fs-fuse et goofys sont tous deux des outils FUSE (système de fichiers dans l'espace utilisateur) qui vous permettent de monter un compartiment Amazon S3 en tant que système de fichiers local sur un cluster. L'outil `s3fs` fournit une expérience similaire à POSIX. L'outil `goofys` est un bon choix lorsque vous préférez les performances à un système de fichiers compatible POSIX.

  La série Amazon EMR 7.x utilise Amazon Linux 2023, qui ne prend pas en charge les référentiels EPEL. Si vous utilisez Amazon EMR 7.x, suivez les instructions d'installation de GitHubs3fs-fuse. `s3fs-fuse` Si vous utilisez les séries 5.x ou 6.x, utilisez les commandes suivantes pour effectuer l'installation. `s3fs-fuse`

  ```
  #!/bin/sh
  ```

```
# Install the s3fs dependency for EMR Notebooks magics
sudo amazon-linux-extras install epel -y
sudo yum install s3fs-fuse -y
```

OU

```
#!/bin/sh

# Install the goofys dependency for EMR Notebooks magics
sudo wget https://github.com/kahing/goofys/releases/latest/download/goofys -P /usr/
bin/
sudo chmod ugo+x /usr/bin/goofys
```

Installer les magics des blocs-notes EMR

> **Note**
>
> Avec les versions 6.0 à 6.9.0 et 5.0 à 5.36.0 d'Amazon EMR, seules les versions 0.2.0 `emr-notebooks-magics` et supérieures du package sont compatibles avec la magic `%mount_workspace_dir`.

Pour installer les magics des blocs-notes EMR, complétez les étapes suivantes.

1. Dans votre bloc-notes, exécutez les commandes suivantes pour installer le package `emr-notebooks-magics`.

   ```
   %pip install boto3 --upgrade
   %pip install botocore --upgrade
   %pip install emr-notebooks-magics --upgrade
   ```

2. Redémarrez votre noyau pour charger les magics des blocs-notes EMR.
3. Vérifiez votre installation à l'aide de la commande suivante, qui devrait afficher le texte d'aide de sortie pour `%mount_workspace_dir`.

   ```
   %mount_workspace_dir?
   ```

Montez un répertoire Workspace avec `%mount_workspace_dir`

La magic `%mount_workspace_dir` vous permet de monter votre répertoire d'espace de travail sur votre cluster EMR afin de pouvoir importer et exécuter d'autres fichiers, modules ou packages stockés dans votre répertoire.

L'exemple suivant monte l'intégralité du répertoire Workspace sur un cluster et spécifie l'argument facultatif *<--fuse-type>* qui permet d'utiliser goofys pour monter le répertoire.

```
%mount_workspace_dir . <--fuse-type goofys>
```

Pour vérifier que votre répertoire Workspace est monté, utilisez l'exemple suivant pour afficher le répertoire de travail actuel avec la commande `ls`. La sortie doit afficher tous les fichiers de votre Workspace.

```
%%sh
ls
```

Lorsque vous avez terminé d'apporter des modifications à votre Workspace, vous pouvez démonter le répertoire du Workspace à l'aide de la commande suivante :

> **Note**
>
> Le répertoire de votre Workspace reste monté sur votre cluster même lorsque le Workspace est arrêté ou détaché. Vous devez démonter explicitement votre répertoire Workspace.

```
%umount_workspace_dir
```

Télécharger un objet Amazon S3 avec `%generate_s3_download_url`

La commande `generate_s3_download_url` crée une URL présignée pour un objet stocké dans Amazon S3. Vous pouvez utiliser l'URL présignée pour télécharger l'objet sur votre ordinateur local. Par exemple, vous pouvez exécuter `generate_s3_download_url` pour télécharger le résultat d'une requête SQL que votre code écrit sur Amazon S3.

L'URL présignée est valide pendant 60 minutes par défaut. Vous pouvez modifier le délai d'expiration en spécifiant un nombre de secondes pour le drapeau `--expires-in`. Par exemple, `--expires-in 1800` crée une URL valide pendant 30 minutes.

L'exemple suivant génère un lien de téléchargement pour un objet en spécifiant le chemin complet d'Amazon S3 : *s3://EXAMPLE-DOC-BUCKET/path/to/my/object*.

```
%generate_s3_download_url s3://EXAMPLE-DOC-BUCKET/path/to/my/object
```

Pour en savoir plus sur l'utilisation de `generate_s3_download_url`, exécutez la commande suivante pour afficher le texte d'aide.

```
%generate_s3_download_url?
```

Exécuter un bloc-notes en mode sans tête avec **`%execute_notebook`**

Grâce à la magic `%execute_notebook`, vous pouvez exécuter un autre bloc-notes en mode sans tête et afficher le résultat de chaque cellule que vous avez exécutée. Cette magic nécessite des autorisations supplémentaires pour le rôle d'instance partagé par Amazon EMR et Amazon EC2. Pour plus de détails sur la façon d'accorder des autorisations supplémentaires, exécutez la commande`%execute_notebook?`.

Au cours d'un travail de longue durée, votre système peut se mettre en veille pour cause d'inactivité ou perdre temporairement la connexion Internet. Cela peut perturber la connexion entre votre navigateur et le serveur Jupyter. Dans ce cas, vous risquez de perdre le résultat des cellules que vous avez exécutées et envoyées depuis le serveur Jupyter.

Si vous utilisez le bloc-notes en mode sans tête avec la magic `%execute_notebook`, le bloc-notes EMR capture les données des cellules qui ont fonctionné, même si le réseau local est perturbé. EMR Notebooks enregistre la sortie de manière incrémentielle dans un nouveau bloc-notes portant le même nom que le bloc-notes que vous avez utilisé. EMR Notebooks place ensuite le bloc-notes dans un nouveau dossier au sein du Workspace. Les exécutions sans tête se produisent sur le même cluster et utilisent le rôle de service `EMR_Notebook_DefaultRole`, mais des arguments supplémentaires peuvent modifier les valeurs par défaut.

Pour exécuter un bloc-notes en mode sans tête, utilisez la commande suivante :

```
%execute_notebook <relative-file-path>
```

Pour spécifier un ID de cluster et un rôle de service pour une exécution en mode sans tête, utilisez la commande suivante :

```
%execute_notebook <notebook_name>.ipynb --cluster-id <emr-cluster-id> --service-role
 <emr-notebook-service-role>
```

Lorsqu'Amazon EMR et Amazon EC2 partagent un rôle d'instance, ce rôle nécessite les autorisations supplémentaires suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:StartNotebookExecution",
                "elasticmapreduce:DescribeNotebookExecution",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<AccoundId>:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

> **Note**
>
> Pour utiliser la magic `%execute_notebook`, installez le package `emr-notebooks-magics`, version 0.2.3 ou supérieure.

Utiliser des blocs-notes multilingues avec des noyaux Spark

Chaque noyau de bloc-notes Jupyter possède une langue par défaut. Par exemple, le langage par défaut du noyau Spark est Scala, et le langage par PySpark défaut du noyau est Python. Avec Amazon EMR 6.4.0 et versions ultérieures, EMR Studio prend en charge les blocs-notes multilingues.

Cela signifie que, en plus de la langue par défaut, chaque noyau d'EMR Studio peut prendre en charge les langages suivants : Python, Spark, R et Spark SQL.

Pour activer cette fonctionnalité, spécifiez l'une des commandes magic suivantes au début de chaque cellule.

Langue	Command
Python	`%%pyspark`
Scala	`%%scalaspark`
R	`%%rspark` Non prise en charge pour les charges de travail interactives avec EMR sans serveur.
SQL Spark	`%%sql`

Lorsqu'elles sont invoquées, ces commandes exécutent la cellule entière au sein de la même session Spark à l'aide de l'interprète de la langue correspondante.

La `%%pyspark` cellule magic permet aux utilisateurs d'écrire PySpark du code dans tous les noyaux Spark.

```
%%pyspark
a = 1
```

La magic cellulaire `%%sql` permet aux utilisateurs d'écrire du code Spark-SQL dans tous les noyaux Spark.

```
%%sql
SHOW TABLES
```

La magic cellulaire `%%rspark` permet aux utilisateurs d'écrire du code SparkR dans tous les noyaux Spark.

```
%%rspark
```

```
a <- 1
```

La magic cellulaire `%%scalaspark` permet aux utilisateurs d'écrire du code Spark Scala dans tous les noyaux Spark.

```
%%scalaspark
val a = 1
```

Partagez les données entre les interprètes linguistiques à l'aide de tableaux temporaires

Vous pouvez également partager des données entre les interprètes linguistiques à l'aide de tableaux temporaires. L'exemple suivant utilise `%%pyspark` dans une cellule pour créer un tableau temporaire en Python et utilise `%%scalaspark` la cellule suivante pour lire les données de ce tableau dans Scala.

```
%%pyspark
df=spark.sql("SELECT * from nyc_top_trips_report LIMIT 20")
# create a temporary table called nyc_top_trips_report_view in python
df.createOrReplaceTempView("nyc_top_trips_report_view")
```

```
%%scalaspark
// read the temp table in scala
val df=spark.sql("SELECT * from nyc_top_trips_report_view")
df.show(5)
```

Présentation des blocs-notes Amazon EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Vous pouvez utiliser Amazon EMR Notebooks ainsi que des clusters Amazon EMR exécutant Apache Spark pour créer et ouvrir Jupyter Notebook et des JupyterLab interfaces au sein de la console Amazon EMR. Le bloc-notes EMR est un bloc-notes « sans serveur » que vous pouvez utiliser pour exécuter des requêtes et du code. Contrairement à un bloc-notes traditionnel, le contenu d'un bloc-notes EMR, c'est-à-dire les équations, les requêtes, les modèles, le code et le texte narratif contenus dans les cellules du bloc-notes, s'exécute 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 plus grande durabilité et une réutilisation plus souple.

Vous pouvez démarrer un cluster, y attacher un bloc-notes EMR pour analyse, puis mettre fin au cluster. Vous pouvez également fermer un bloc-notes relié à un cluster en cours d'exécution et basculer vers un autre. Plusieurs utilisateurs peuvent attacher simultanément des blocs-notes au même cluster et partager entre eux des fichiers de blocs-notes sur Amazon S3. 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 par programmation à l'aide de l'API Amazon EMR, sans avoir à interagir avec la console Amazon EMR (« exécution sans tête »). Vous devez inclure une cellule dans le bloc-notes EMR contenant une balise de paramètres. Cette cellule permet à un script de transmettre de nouvelles valeurs d'entrée au bloc-notes. Les blocs-notes paramétrés peuvent être réutilisés avec différents ensembles de valeurs d'entrée. Il n'est pas nécessaire de faire des copies du même bloc-notes pour le modifier et l'exécuter avec de nouvelles valeurs d'entrée. Amazon EMR crée et enregistre le bloc-notes résultant sur S3 pour chaque exécution du bloc-

notes paramétré. Pour des exemples de code d'API de bloc-notes EMR, consultez Exemples de commandes pour l'exécution de blocs-notes EMR par programmation.

> **Important**
>
> La fonctionnalité de blocs-notes EMR prend en charge les clusters qui utilisent les versions 5.18.0 et supérieures d'Amazon EMR. Nous vous recommandons d'utiliser les blocs-notes EMR avec des clusters qui utilisent la dernière version d'Amazon EMR, ou au moins 5.30.0, 5.32.0 ou 6.2.0. Avec ces versions, les noyaux Jupyter s'exécutent sur le cluster attaché plutôt que sur une instance Jupyter. Cela améliore les performances et votre capacité à personnaliser les noyaux et les bibliothèques. Pour plus d'informations, consultez Différences de capacités en fonction de la version du cluster.

Les frais afférents au stockage Amazon S3 et aux clusters Amazon EMR s'appliquent.

Les notebooks Amazon EMR sont disponibles sous forme d'espaces de travail Amazon EMR Studio dans la console

Transition des blocs-notes EMR vers les espaces de travail

Dans la nouvelle console Amazon EMR, nous avons fusionné les blocs-notes EMR et les espaces de travail Amazon EMR Studio en une seule expérience. Lorsque vous utilisez un studio EMR, vous pouvez créer et configurer différents espaces de travail pour organiser et exécuter des blocs-notes. Si vous disposiez de blocs-notes Amazon EMR dans l'ancienne console, ils sont disponibles sous forme d'espaces de travail EMR Studio dans la nouvelle console.

Amazon EMR a créé ces nouveaux espaces de travail EMR Studio pour vous. Le nombre de studios que nous avons créés correspond au nombre de VPC distincts que vous utilisez à partir des blocs-notes EMR. Par exemple, si vous vous connectez à des clusters EMR dans deux VPC différents à partir de blocs-notes EMR, nous avons créé deux nouveaux studios EMR. Vos blocs-notes sont répartis entre les nouveaux studios.

> **Important**
>
> Nous avons désactivé l'option permettant de créer de nouveaux blocs-notes dans l'ancienne console Amazon EMR. Utilisez plutôt Créer un espace de travail dans la nouvelle console Amazon EMR.

Pour plus d'informations sur les espaces de travail Amazon EMR Studio, consultez Découvrir les bases de l'espace de Workspace. Pour une présentation conceptuelle d'EMR Studio, consultez Espaces de travail sur la page Comment fonctionne Amazon EMR Studio.

Que devez-vous faire ?

Bien que vous puissiez toujours utiliser vos blocs-notes existants dans l'ancienne console, nous vous recommandons d'utiliser les espaces de travail Amazon EMR Studio dans la nouvelle console. Vous devez configurer des autorisations de rôle supplémentaires pour activer les fonctionnalités d'EMR Studio qui ne sont pas disponibles dans les blocs-notes EMR.

> **Note**
>
> Au minimum, pour visualiser les blocs-notes EMR existants en tant qu'espaces de travail EMR Studio et pour créer de nouveaux espaces de travail, les utilisateurs doivent disposer des autorisations `elasticmapreduce:ListStudios` et `elasticmapreduce:CreateStudioPresignedUrl` sur leurs rôles. Pour accéder à toutes les fonctionnalités d'EMR Studio, consultez Activation des fonctionnalités d'EMR Studio pour les utilisateurs des blocs-notes EMR pour la liste complète des autorisations supplémentaires dont les utilisateurs des blocs-notes EMR auront besoin.

Fonctionnalités améliorées d'EMR Studio au-delà des blocs-notes EMR

Grâce à Amazon EMR Studio, vous pouvez configurer et utiliser les fonctionnalités suivantes, qui ne sont pas disponibles avec les blocs-notes EMR :

- Navigation et attachement aux clusters EMR à partir de Jupyterlab
- Navigation et attachement aux clusters virtuels des blocs-notes EMR à partir de Jupyterlab
- Connexion aux dépôts Git depuis Jupyterlab
- Collaboration avec d'autres membres de votre équipe pour écrire et exécuter le code du bloc-notes

- Consultation des données à l'aide de SQL Explorer
- Provisionnement des clusters EMR avec Service Catalog

Pour une liste complète des fonctionnalités d'Amazon EMR Studio, consultez Principales fonctionnalités d'EMR Studio.

Activation des fonctionnalités d'EMR Studio pour les utilisateurs des blocs-notes EMR

Les nouveaux studios EMR que nous allons créer dans le cadre de cette fusion utilisent le rôle IAM `EMR_Notebooks_DefaultRole` existant comme fonction du service EMR Studio.

Les utilisateurs qui font la transition des blocs-notes EMR vers EMR Studio et qui souhaitent utiliser les capacités supplémentaires d'EMR Studio ont besoin de plusieurs nouvelles autorisations de rôle. Ajoutez les autorisations suivantes aux rôles des utilisateurs de vos blocs-notes EMR qui prévoient d'utiliser EMR Studio.

> **Note**
>
> Au minimum, pour visualiser les blocs-notes EMR existants en tant qu'espaces de travail EMR Studio et pour créer de nouveaux espaces de travail, les utilisateurs doivent disposer des autorisations `elasticmapreduce:ListStudios` et `elasticmapreduce:CreateStudioPresignedUrl` sur leurs rôles. Pour utiliser toutes les fonctionnalités d'EMR Studio, ajoutez toutes les autorisations répertoriées ci-dessous. Les utilisateurs administrateurs doivent également être autorisés à créer et à gérer un studio EMR. Pour plus d'informations, consultez Autorisations d'administrateur pour créer et gérer un EMR Studio.

```
"elasticmapreduce:DescribeStudio",
"elasticmapreduce:ListStudios",
"elasticmapreduce:CreateStudioPresignedUrl",
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce:DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities",
"emr-containers:ListVirtualClusters",
"emr-containers:DescribeVirtualCluster",
```

```
"emr-containers:ListManagedEndpoints",
"emr-containers:DescribeManagedEndpoint",
"emr-containers:CreateAccessTokenForManagedEndpoint",
"emr-containers:ListJobRuns",
"emr-containers:DescribeJobRun",
"servicecatalog:SearchProducts",
"servicecatalog:DescribeProduct",
"servicecatalog:DescribeProductView",
"servicecatalog:DescribeProvisioningParameters",
"servicecatalog:ProvisionProduct",
"servicecatalog:UpdateProvisionedProduct",
"servicecatalog:ListProvisioningArtifacts",
"servicecatalog:DescribeRecord",
"servicecatalog:ListLaunchPaths",
"cloudformation:DescribeStackResources"
```

Les autorisations suivantes sont également nécessaires pour utiliser les fonctionnalités de collaboration d'EMR Studio, mais ne l'étaient pas pour les blocs-notes EMR.

```
"sso-directory:SearchUsers",
"iam:GetUser",
"iam:GetRole",
"iam:ListUsers",
"iam:ListRoles",
"sso:GetManagedApplicationInstance"
```

Considérations relatives à l'utilisation des blocs-notes EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

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

Exigences en matière de cluster

- Activer le blocage de l'accès public à Amazon EMR – L'accès entrant à un cluster permet aux utilisateurs du cluster d'exécuter des noyaux de bloc-notes. 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, consultez Utilisation du blocage de l'accès public Amazon EMR et Contrôle du trafic réseau avec des groupes de sécurité.
- Utiliser un cluster compatible – Un cluster attaché à un bloc-notes doit répondre aux exigences suivantes :
  - Seuls les clusters créés à l'aide d'Amazon EMR sont pris en charge. Vous pouvez créer un cluster indépendamment à l'intérieur d'Amazon EMR, puis attacher un bloc-notes EMR, ou vous pouvez créer un cluster compatible lorsque vous créez un bloc-notes EMR.
  - Seuls les clusters créés à l'aide d'Amazon EMR en version 5.18.0 et ultérieure sont pris en charge. veuillez consulter the section called “Différences de capacités en fonction de la version du cluster”.
  - Les clusters créés à l'aide d'instances Amazon EC2 avec des processeurs AMD EPYC, par exemple, les types d'instances m5a.* et r5a.*, ne sont pas pris en charge.
  - Les blocs-notes EMR ne fonctionnent qu'avec des clusters créés avec `VisibleToAllUsers` défini sur `true`. `VisibleToAllUsers` est `true` par défaut.
  - 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 les blocs-notes EMR ne prennent actuellement en charge que les clusters Spark.

    > **Important**
    >
    > Pour les versions 5.32.0 et ultérieures ou 6.2.0 et ultérieures d'Amazon EMR, votre cluster doit également exécuter l'application Jupyter Enterprise Gateway pour pouvoir utiliser les blocs-notes EMR.

  - Les clusters avec authentification Kerberos ne sont pas pris en charge.
  - Les clusters intégrés AWS Lake Formation prennent en charge l'installation de bibliothèques adaptées aux ordinateurs portables uniquement. L'installation des noyaux et des bibliothèques sur le cluster n'est pas prise en charge.

- Les clusters avec plusieurs nœuds primaires ne sont pas pris en charge.
- Les clusters utilisant des instances Amazon EC2 basées sur AWS Graviton2 ne sont pas pris en charge.

Différences de capacités en fonction de la version du cluster

Nous vous recommandons vivement d'utiliser les blocs-notes EMR avec les clusters créés à l'aide des versions 5.30.0, 5.32.0 ou ultérieures, ou 6.2.0 ou ultérieures d'Amazon EMR. Avec ces versions, les blocs-notes EMR exécutent les noyaux sur le cluster Amazon EMR attaché. Les noyaux et les bibliothèques peuvent être installés directement sur le nœud primaire du cluster. L'utilisation des blocs-notes EMR avec ces versions de cluster présente les avantages suivants :

- Performances améliorées – Les noyaux de bloc-notes s'exécutent sur des clusters avec les types d'instance 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 noyaux – Vous pouvez vous connecter au cluster pour installer des paquets de noyau en utilisant `conda` et `pip`. En outre, l'installation `pip` est prise en charge à l'aide de commandes de terminal dans les cellules de bloc-notes. Dans les versions précédentes, seuls les noyaux préinstallés étaient disponibles (Python PySpark, Spark et SparkR). Pour plus d'informations, consultez Installation des noyaux et des bibliothèques Python sur le nœud primaire d'un cluster.
- Possibilité d'installer des bibliothèques Python – Vous pouvez installer des bibliothèques Python sur le nœud primaire du cluster en utilisant `conda` et `pip`. Nous vous recommandons d'utiliser `conda`. Dans les versions antérieures, seules les bibliothèques adaptées aux ordinateurs portables sont prises en charge. PySpark

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

Version de cluster	Bibliothèques adaptées aux ordinateurs portables pour PySpark	Installation du noyau sur le cluster	Installation de la bibliothèque Python sur le nœud primaire
Antérieur à 5.18.0	Blocs-notes EMR non pris en charge		
5.18.0–5.25.0	Non	Non	Non

Version de cluster	Bibliothèques adaptées aux ordinateurs portables pour PySpark	Installation du noyau sur le cluster	Installation de la bibliothèque Python sur le nœud primaire
5.26.0–5.29.0	Oui	Non	Non
5.30.0	Oui	Oui	Oui
6.0.0	Non	Non	Non
Version 5.32.0 et ultérieure, et version 6.2.0 et ultérieure	Oui	Oui	Oui

Limites pour les blocs-notes EMR connectés simultanément

Prenez en compte le type d'instance EC2 du nœud primaire 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 du nœud primaire	Nombre de blocs-notes EMR
*.medium	2
*.large	4
*.xlarge	8
*.2xlarge	16
*.4xlarge	24
*.8xlarge	24
*.16xlarge	24

Versions de bloc-notes Jupyter et de Python

Les blocs-notes EMR exécutent le bloc-notes Jupyter en version 6.0.2 et Python en version 3.6.5, quelle que soit la version Amazon EMR du cluster attaché.

Considérations relatives à la sécurité

Utiliser des emplacements S3 chiffrés

Si vous indiquez un emplacement chiffré dans Amazon S3 pour stocker les fichiers de bloc-notes, vous devez configurer le rôle Rôle de service pour Blocs-notes EMR en tant qu'utilisateur de clé. Le rôle de service par défaut est `EMR_Notebooks_DefaultRole`. Si vous utilisez une AWS KMS clé pour le chiffrement, consultez la section Utilisation des politiques relatives aux clés dans AWS KMS dans le manuel du AWS Key Management Service développeur et l'article d'assistance relatif à l'ajout d'utilisateurs clés.

Utilisation de cookies dans les domaines d’hébergement

Pour renforcer la sécurité des applications hors console que vous pouvez utiliser avec Amazon EMR, les domaines hébergeant les applications sont enregistrés dans la liste des suffixes publics (PSL). Voici des exemples de ces domaines d’hébergement : `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Pour plus de sécurité, si vous avez besoin de définir des cookies sensibles dans le nom de domaine par défaut, nous vous recommandons d’utiliser des cookies avec un préfixe `__Host-`. Cela vous permettra de protéger votre domaine contre les tentatives de falsification de requêtes intersites (CSRF). Pour plus d’informations, voir la page Set-Cookie du Mozilla Developer Network.

Création d'un bloc-notes

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Créez un bloc-notes EMR à l'aide de l'ancienne console Amazon EMR. La création de blocs-notes à l'aide de l'API Amazon EMR AWS CLI ou de l'API Amazon EMR n'est pas prise en charge.

Pour créer un bloc-notes EMR

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Notebooks (Blocs-notes), Create notebook (Créer bloc-notes).
3. Saisissez un nom Notebook name (Nom de bloc-notes) et une Notebook description (Description de bloc-notes) facultative.
4. Si vous disposez d'un cluster actif auquel vous souhaitez attacher le bloc-notes, laissez la valeur par défaut Choisir un cluster existant sélectionnée, cliquez sur Choisir, sélectionnez un cluster dans la liste, puis cliquez sur Choisir un cluster. Pour plus d'informations sur les exigences en matière de cluster pour les blocs-notes EMR, consultez Considérations relatives à l'utilisation des blocs-notes EMR.

   —ou—

   Choisissez Créer un cluster, entrez un nom de cluster et choisissez les options selon les instructions 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. La valeur par défaut est la dernière version d'Amazon EMR (5.36.2).
Applications	Impossible de modifier. Répertorie les applications installées sur le cluster.
Instance	Entrez le nombre d'instances et sélection nez le type d'instance EC2. Une instance est utilisée pour le nœud primaire. Les autres sont utilisées pour les nœuds principaux. Le type d'instance détermine le nombre de blocs-notes pouvant être attachés au cluster

Paramètre	Description
	simultanément. Pour plus d’informations, consultez Limites pour les blocs-notes EMR connectés simultanément.
Rôle EMR	Laissez la valeur par défaut ou cliquez sur le lien pour spécifier une fonction de service personnalisée pour Amazon EMR. Pour plus d’informations, consultez Rôle de service pour Amazon EMR (rôle EMR).
Profil d'instance EC2	Laissez la valeur par défaut ou choisisse z le lien pour spécifier un rôle de service personnalisé pour les instances EC2. Pour plus d’informations, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).
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 primaire à l'aide de SSH.
Arrêt automatique	L'arrêt automatique est pris en charge pour les versions 5.30.0 et 6.1.0 et ultérieures d'Amazon EMR. Cochez la case pour activer l'arrêt automatiq ue, puis indiquez la durée d'inactivité après laquelle le cluster doit s'arrêter automatiq uement. Pour plus d’informations, consultez Utilisation d'une politique de résiliation automatique.

5. Pour Security groups (Groupes de sécurité), choisissez Use default security groups (Utiliser les groupes de sécurité par défaut). Vous pouvez également sélectionner Choisir des groupes de sécurité et sélectionner des groupes de sécurité personnalisés qui sont disponibles dans le VPC

du cluster. Sélectionnez un pour l'instance principale et un autre pour l'instance client du bloc-notes. Pour plus d'informations, consultez the section called "Groupes de sécurité pour Blocs-notes EMR".

6. Pour la fonction du 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 Rôle de service pour Blocs-notes EMR.
7. Pour l'emplacement du bloc-notes, choisissez l'emplacement dans Amazon S3 où le fichier du bloc-notes est enregistré, ou indiquez votre propre emplacement. Si le compartiment et le dossier n'existent pas, Amazon EMR le crée.

   Amazon EMR crée un dossier avec l'identifiant du bloc-notes comme nom de dossier, et enregistre le bloc-notes dans un fichier nommé *`NotebookName`*`.ipynb`. Par exemple, si vous indiquez l'emplacement `s3://MyBucket/MyNotebooks` dans Amazon S3 pour un bloc-notes nommé `MyFirstEMRManagedNotebook`, le fichier est enregistré sous `s3://MyBucket/MyNotebooks/`*`NotebookID`*`/MyFirstEMRManagedNotebook.ipynb`.

   Si vous indiquez un emplacement chiffré dans Amazon S3, vous devez configurer le rôle Rôle de service pour Blocs-notes EMR en tant qu'utilisateur de clé. Le rôle de service par défaut est `EMR_Notebooks_DefaultRole`. Si vous utilisez une AWS KMS clé pour le chiffrement, consultez la section Utilisation des politiques relatives aux clés dans AWS KMS dans le manuel du AWS Key Management Service développeur et l'article d'assistance relatif à l'ajout d'utilisateurs clés.
8. En option, si vous avez ajouté à Amazon EMR un référentiel basé sur Git que vous souhaitez l'associer à ce bloc-notes, choisissez Référentiel Git, sélectionnez Choisir un référentiel, puis sélectionnez un référentiel dans la liste. Pour plus d'informations, consultez – Association de référentiels Git à des blocs-notes EMR.
9. Le cas échéant, choisissez Tags (Balises), puis ajoutez des balises clé-valeur supplémentaires pour le bloc-notes.

   > **Important**
   >
   > Une balise par défaut avec l'ensemble de chaîne de Key (Clé) définie sur `creatorUserID` et la valeur définie sur votre ID d'utilisateur IAM sont appliqués 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 politiques IAM de contrôle d'accès.

10. Choisissez Créer un bloc-notes.

Utilisation des blocs-notes EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Après la création d'un bloc-notes EMR, le démarrage du bloc-notes prend peu de temps. Le Status (Statut) dans la liste Blocs-notes affiche Démarrage. Vous pouvez ouvrir un bloc-notes lorsqu'il est Ready (Prêt). Un bloc-notes peut prendre un peu plus longtemps à être Ready (Prêt) si vous avez créé un cluster pour l'accompagner.

> **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.

Compréhension de l'état du bloc-notes

Le bloc-notes EMR peut avoir les états suivants dans la liste des blocs-notes.

Statut	Signification
Prêt	Vous pouvez ouvrir le bloc-notes à l'aide de l'éditeur de bloc-notes. Vous pouvez arrêter ou supprimer un bloc-notes, même s'il affiche

Statut	Signification
	un statut de Ready (Prêt). Vous devez d'abord arrêter le bloc-notes pour modifier les clusters. Si un bloc-notes avec un statut de Ready (Prêt) est inactif pendant une longue période de temps, il est automatiquement arrêté.
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 attente	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 bloc-notes avec le bloc-notes en mode local. Tout code qui s'appuie sur les processus de cluster n'est pas exécuté et échoue.
Arrêt en cours	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.

Statut	Signification
Suppression	Le cluster est en cours de suppression de la liste des clusters disponibles. Le fichier de bloc-notes *NotebookName* `.ipynb` reste dans Amazon S3 et continue à accumuler des frais de stockage applicables.

Utilisation de l'éditeur de bloc-notes

L'un des avantages de l'utilisation d'un bloc-notes EMR est que vous pouvez lancer le bloc-notes dans Jupyter ou JupyterLab directement depuis la console.

Avec EMR Notebooks, l'éditeur de bloc-notes auquel vous accédez depuis la console Amazon EMR est l'éditeur open source Jupyter Notebooks ou. JupyterLab L'éditeur de bloc-notes étant lancé dans la console Amazon EMR, il est plus efficace de configurer l'accès qu'avec un bloc-notes hébergé sur un cluster Amazon EMR. 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 lui suffit d'ouvrir l'éditeur de bloc-notes dans la console Amazon EMR.

Seul un utilisateur à la fois peut avoir un bloc-notes EMR ouvert à partir d'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 de l'éditeur de bloc-notes, qui n'est valable que pour une courte durée. 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 bloc-notes, accordez des autorisations à leur utilisateur via des politiques d'autorisation et assurez-vous que la fonction de service associée aux blocs-notes EMR a accès à l'emplacement Amazon S3. Pour plus d'informations, consultez the section called “Sécurité” et Rôle de service pour Blocs-notes EMR.

Ouverture de l'éditeur de bloc-notes pour un bloc-notes EMR

1. Sélectionnez un bloc-notes avec un Status (Statut) de Ready (Prêt) ou Pending (En attente) à partir de la liste Notebooks (Blocs-notes).
2. Choisissez Ouvrir dans JupyterLab ou Ouvrir dans Jupyter.

   Un nouvel onglet de navigateur s'ouvre dans l'éditeur JupyterLab Jupyter Notebook.
3. Depuis le menu Kernel (Noyau), choisissez Change kernel (Changer de noyau) puis sélectionnez le noyau pour votre langage 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 du bloc-notes et les résultats sont enregistrés automatiquement dans le fichier du bloc-notes et périodiquement sur Amazon S3. Un bloc-notes dans lequel aucune modification n'a été apportée depuis la dernière fois qu'une cellule a été modifiée affiche (autosaved) (enregistré automatiquement) à côté du nom de bloc-notes dans l'éditeur. Si des modifications n'ont pas encore été enregistrées, unsaved changes (modifications non enregistrées) s'affiche.

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

Modification des clusters

Vous pouvez modifier le cluster auquel un bloc-notes EMR est attaché sans modifier le contenu du bloc-notes lui-même. Vous pouvez modifier les clusters pour les seuls blocs-notes qui ont un état Stopped (Arrêté).

Modification du cluster d'un bloc-notes EMR

1. Si le bloc-notes que vous souhaitez modifier est en cours d'exécution, sélectionnez-le dans la liste Notebooks (Blocs-notes) et choisissez Arrêter.
2. Lorsque l'état de bloc-notes est Stopped (Arrêté), sélectionnez le bloc-notes dans la liste Notebooks (Blocs-notes), puis choisissez View details (Afficher les détails).

3. Choisissez Change cluster (Changer de 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—

   Choisissez Create a cluster (Créer un cluster), puis choisissez les options de cluster. Pour plus d'informations, consultez Exigences en matière de cluster.
5. Choisissez une option pour les Security groups (Groupes de sécurité), puis choisissez Modifier le cluster et démarrez le bloc-notes.

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

Lorsque vous supprimez un bloc-notes EMR aide de la console Amazon EMR, vous devez supprimer le bloc-notes à partir de la liste des blocs-notes disponibles. Cependant, les fichiers de bloc-notes restent dans Amazon S3 et continuent d'accumuler des frais de stockage.

Pour supprimer un bloc-notes et retirer les fichiers associés

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Notebooks (Blocs-notes), sélectionnez votre bloc-notes dans la liste, puis choisissez View details (Afficher les détails).
3. Choisissez l'icône de dossier à côté de Notebook location (Emplacement de bloc-notes) et copiez l'URL qui se trouve dans le modèle `s3://`*`MyNotebookLocationPath`*`/`*`NotebookID`*`/`.
4. Sélectionnez Delete (Supprimer).

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

   —ou—

   Si vous l'avez AWS CLI installé, 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 le AWS CLI est configuré avec les clés d'accès d'un

utilisateur autorisé à supprimer l'emplacement Amazon S3. Pour plus d'informations, veuillez consulter configuration de l'outil AWS CLI dans le guide de l'utilisateur de l'outil AWS Command Line Interface .

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

Partage de fichiers de bloc-notes

Chaque bloc-notes EMR est enregistré sur Amazon S3 sous la forme d'un fichier nommé *NotebookName*.ipynb. Tant qu'un fichier de bloc-notes est compatible avec la même version de bloc-notes Jupyter que celle sur laquelle repose le bloc-notes EMR, vous pouvez ouvrir le bloc-notes en tant que bloc-notes EMR.

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

Utilisation d'un autre fichier de bloc-notes comme base d'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. Notez l'emplacement dans Amazon S3 que vous avez choisi 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.

   La AWS CLI commande suivante pour Amazon S3 remplace un fichier enregistré sur une machine locale appelée SharedNotebook.ipynb pour un bloc-notes EMR avec le nom

MyNotebooke-12A3BCDEFJHIJKLMNO45PQRST, l'identifiant et créé avec les informations MyBucket/MyNotebooksFolder spécifiées dans Amazon S3. Pour plus d'informations sur l'utilisation de la console Amazon S3 pour copier et remplacer des fichiers, consultez la rubrique Chargement, téléchargement et gestion d'objets dans le Guide de l'utilisateur d'Amazon Simple Storage Service.

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

Exemples de commandes pour l'exécution de blocs-notes EMR par programmation

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Présentation

Vous pouvez exécuter les blocs-notes EMR à l'aide des API d'exécution à partir d'un script ou de la ligne de commande. Lorsque vous démarrez, arrêtez, listez et décrivez des exécutions de blocs-notes EMR en dehors de la AWS console, vous pouvez contrôler un bloc-notes EMR par programme. Vous pouvez transmettre différentes valeurs de paramètres à un bloc-notes avec une cellule de bloc-notes paramétrée. Il n'est donc plus nécessaire de créer une copie du bloc-notes pour chaque nouvel ensemble de valeurs de paramètres. Pour plus d'informations, consultez la rubrique Actions de l'API Amazon EMR.

Vous pouvez planifier ou regrouper les exécutions de blocs-notes EMR avec Amazon CloudWatch Events et. AWS Lambda Pour plus d'informations, consultez Utilisation AWS Lambda avec Amazon CloudWatch Events.

Autorisations de rôle pour l'exécution par programmation

Pour utiliser l'exécution par programmation avec les blocs-notes EMR, vous devez configurer les autorisations des utilisateurs avec les politiques suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowExecutionActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:StartNotebookExecution",
                "elasticmapreduce:DescribeNotebookExecution",
                "elasticmapreduce:ListNotebookExecutions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowPassingServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

Lorsque vous exécutez par programmation des blocs-notes EMR sur un cluster de blocs-notes EMR, vous devez ajouter ces autorisations supplémentaires :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRetrievingManagedEndpointCredentials",
            "Effect": "Allow",
            "Action": [
                "emr-containers:GetManagedEndpointSessionCredentials"
            ],
            "Resource": [
```

```
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ],
            "Condition": {
                "StringEquals": {
                    "emr-containers:ExecutionRoleArn": [
                        "arn:aws:iam::account-id:role/emr-on-eks-execution-role"
                    ]
                }
            }
        },
        {
            "Sid": "AllowDescribingManagedEndpoint",
            "Effect": "Allow",
            "Action": [
                "emr-containers:DescribeManagedEndpoint"
            ],
            "Resource": [
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ]
        }
    ]
}
```

Limites de l'exécution par programmation

- Un maximum de 100 exécutions simultanées sont prises en charge Région AWS par compte.
- Une exécution est terminée si elle dure plus de 30 jours.
- L'exécution par programmation des blocs-notes n'est pas prise en charge par les applications interactives Amazon EMR sans serveur.

Exemples d'exécution par programmation d'un bloc-notes EMR

Les sections suivantes fournissent plusieurs exemples d'exécution programmatique d'un bloc-notes EMR avec AWS CLI le SDK Boto3 (Python) et Ruby :

- Exemples de commandes CLI pour l'exécution de blocs-notes
- Exemples Python d'exécution d'un bloc-notes
- Exemples Ruby d'exécution de bloc-notes

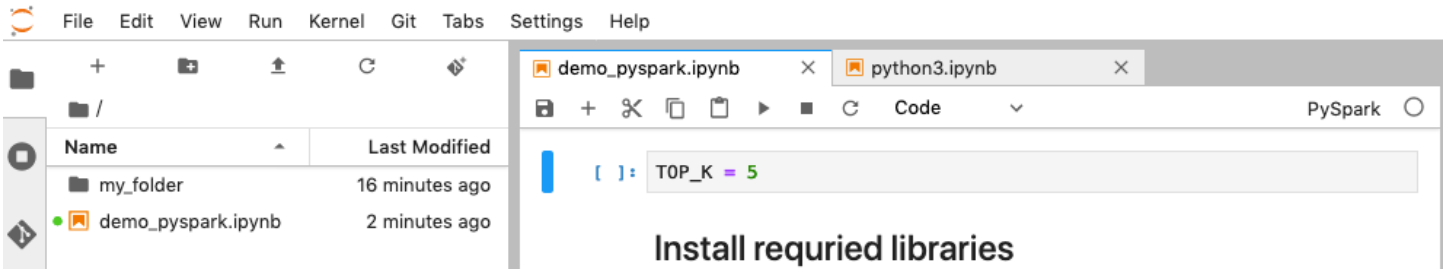
Vous pouvez également exécuter des blocs-notes paramétrés dans le cadre de flux de travail planifiés à l'aide d'un outil d'orchestration tel qu'Apache Airflow ou Amazon Managed Workflows for Apache Airflow (MWAA). Pour plus d'informations, consultez la rubrique Orchestration des tâches d'analyse sur les blocs-notes EMR à l'aide de MWAA sur le blog AWS Big Data.

Exemples de commandes CLI pour l'exécution de blocs-notes

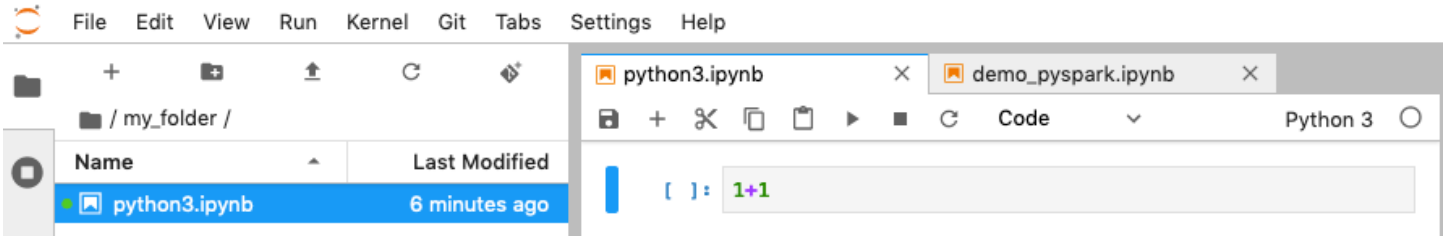
> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

L'exemple suivant utilise le bloc-notes de démonstration de la console des bloc-notes EMR. Pour localiser le bloc-notes, utilisez le chemin relatif du fichier à partir du répertoire de base. Dans cet exemple, vous pouvez exécuter deux fichiers de bloc-notes : `demo_pyspark.ipynb` et`my_folder/python3.ipynb`.

Le chemin relatif du fichier `demo_pyspark.ipynb` est `demo_pyspark.ipynb`, comme indiqué ci-dessous.



Le chemin relatif pour `python3.ipynb` est `my_folder/python3.ipynb`, comme indiqué ci-dessous.



Pour plus d'informations sur les actions `NotebookExecution` de l'API Amazon EMR, consultez la rubrique Actions de l'API Amazon EMR.

Exécution d'un bloc-notes

Vous pouvez utiliser le AWS CLI pour exécuter votre bloc-notes avec l'`start-notebook-execution`action, comme le montrent les exemples suivants.

Example – Exécution d'un bloc-notes EMR dans un espace de travail EMR Studio avec un cluster Amazon EMR (fonctionnant sur Amazon EC2)

```
aws emr --region us-east-1 \
start-notebook-execution \
--editor-id e-ABCDEFG123456 \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"]}'
 \
--relative-path test.ipynb \
--notebook-execution-name my-execution \
--execution-engine '{"Id" : "j-1234ABCD123"}' \
--service-role EMR_Notebooks_DefaultRole

{
    "NotebookExecutionId": "ex-ABCDEFGHIJ1234ABCD"
}
```

Example – Exécution d'un bloc-notes EMR dans un espace de travail EMR Studio avec un cluster de bloc-notes EMR

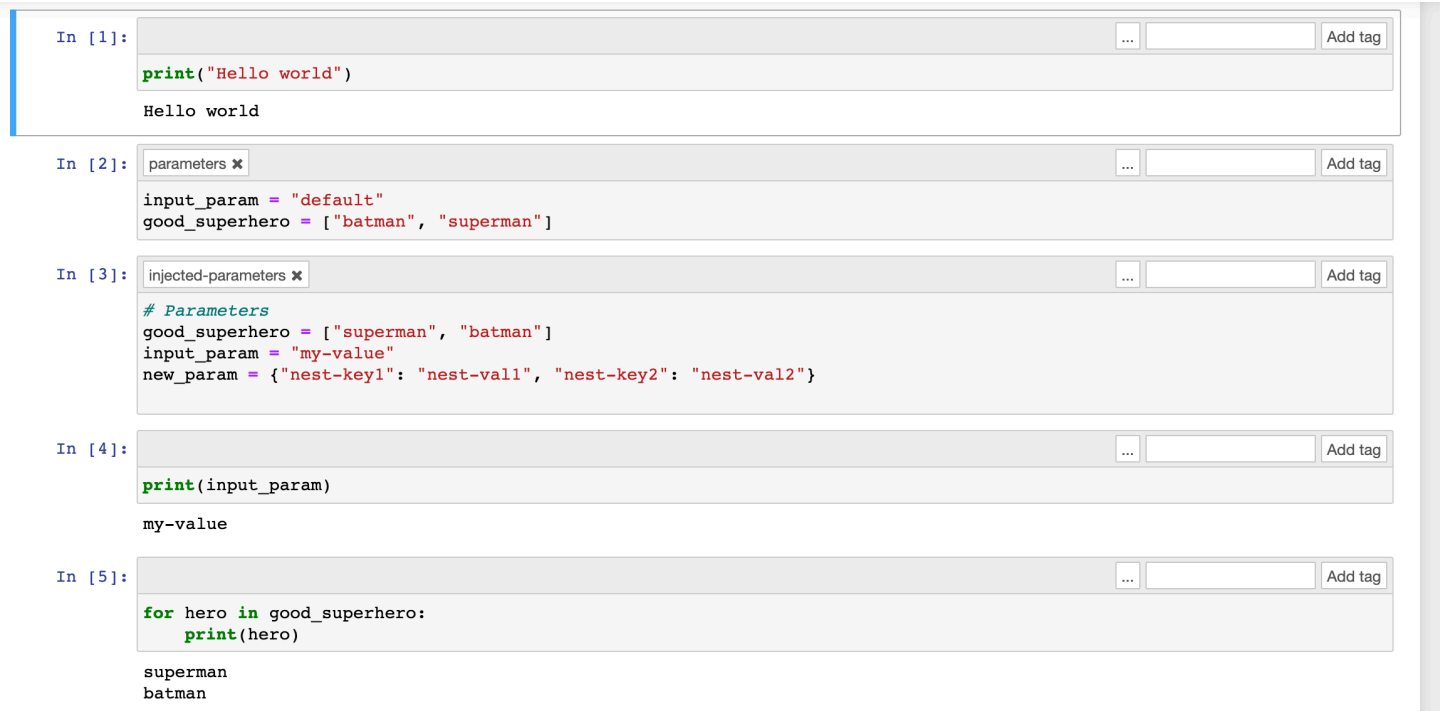
```
aws emr start-notebook-execution \
    --region us-east-1 \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEFG/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --editor-id e-ABCDEFG \
    --relative-path EMRonEKS-spark_python.ipynb
```

Example – Exécution d'un bloc-notes EMR en spécifiant son emplacement Amazon S3

```
aws emr start-notebook-execution \
    --region us-east-1 \
    --notebook-execution-name my-execution-on-emr-on-eks-cluster \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEF/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-to-notebook-
location/EMRonEKS-spark_python.ipynb"}' \
    --output-notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-for-
storing-output-notebook"}'
```

Sortie de bloc-note

Voici le résultat d'un exemple de bloc-notes. La cellule 3 indique les valeurs des paramètres nouvellement injectés.



Décrire un bloc-notes

Vous pouvez utiliser l'action `describe-notebook-execution` pour accéder aux informations relatives à l'exécution d'un bloc-notes spécifique.

```
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": []
    }
}
```

Arrêter un bloc-notes

Si votre bloc-notes exécute une exécution que vous souhaitez arrêter, vous pouvez le faire à l'aide de la commande `stop-notebook-execution`.

```
# 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": []
    }
}
```

Répertorier les exécutions d'un bloc-notes par heure de début

Vous pouvez passer un paramètre `--from` à `list-notebook-executions` pour répertorier les exécutions de votre bloc-notes par heure de début.

```
# 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
        }
    ]
}
```

Répertorier les exécutions d'un bloc-notes par heure de début et par état

La commande `list-notebook-executions` peut également utiliser un paramètre `--status` pour filtrer les résultats.

```
# 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
        }
    ]
}
```

Exemples Python d'exécution d'un bloc-notes

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

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

Pour plus d'informations sur les actions `NotebookExecution` de l'API Amazon EMR, consultez la rubrique Actions de l'API Amazon EMR.

```
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 le résultat de l'exécution `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}}
```

Exemples Ruby d'exécution de bloc-notes

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Voici des exemples de code Ruby illustrant l'utilisation de l'API d'exécution du bloc-notes.

```
# prepare an Amazon 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 bloc-notes et obtention de l'identifiant d'exécution

Dans cet exemple, l'éditeur Amazon S3 et le bloc-notes EMR sont `s3://mybucket/notebooks/e-EA8VGAA429FEQTC8HC9ZHWISK/test.ipynb`.

Pour plus d'informations sur les actions `NotebookExecution` de l'API Amazon EMR, consultez la rubrique Actions de l'API Amazon 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 bloc-notes et impression des détails

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

Le résultat des commandes ci-dessus sera le suivant.

```
{
:notebook_execution_id=>"ex-IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:editor_id=>"e-EA8VGAA429FEQTC8HC9ZHWISK",
:execution_engine=>{:id=>"j-3U82I95AMALGE", :type=>"EMR", :master_instance_security_group_id=>n
: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=>[]
}
```

Filtres de bloc-notes

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

Arrêt de l'exécution du bloc-notes

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

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

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Les blocs-notes Amazon EMR vous permet de configurer l'utilisation de l'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. De plus, les blocs-notes EMR disposent d'un widget de bloc-notes Jupyter intégré pour visualiser les détails des tâches Spark à côté des résultats de la requête dans l'éditeur du 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 primaire.

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

Vous activez cette fonctionnalité en définissant des propriétés dans les classifications de configuration `livy-conf` et `core-site`. 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 de classifications de configuration pour personnaliser des applications, consultez la rubrique Configuration des applications dans le Guide de mise à jour d'Amazon EMR.

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

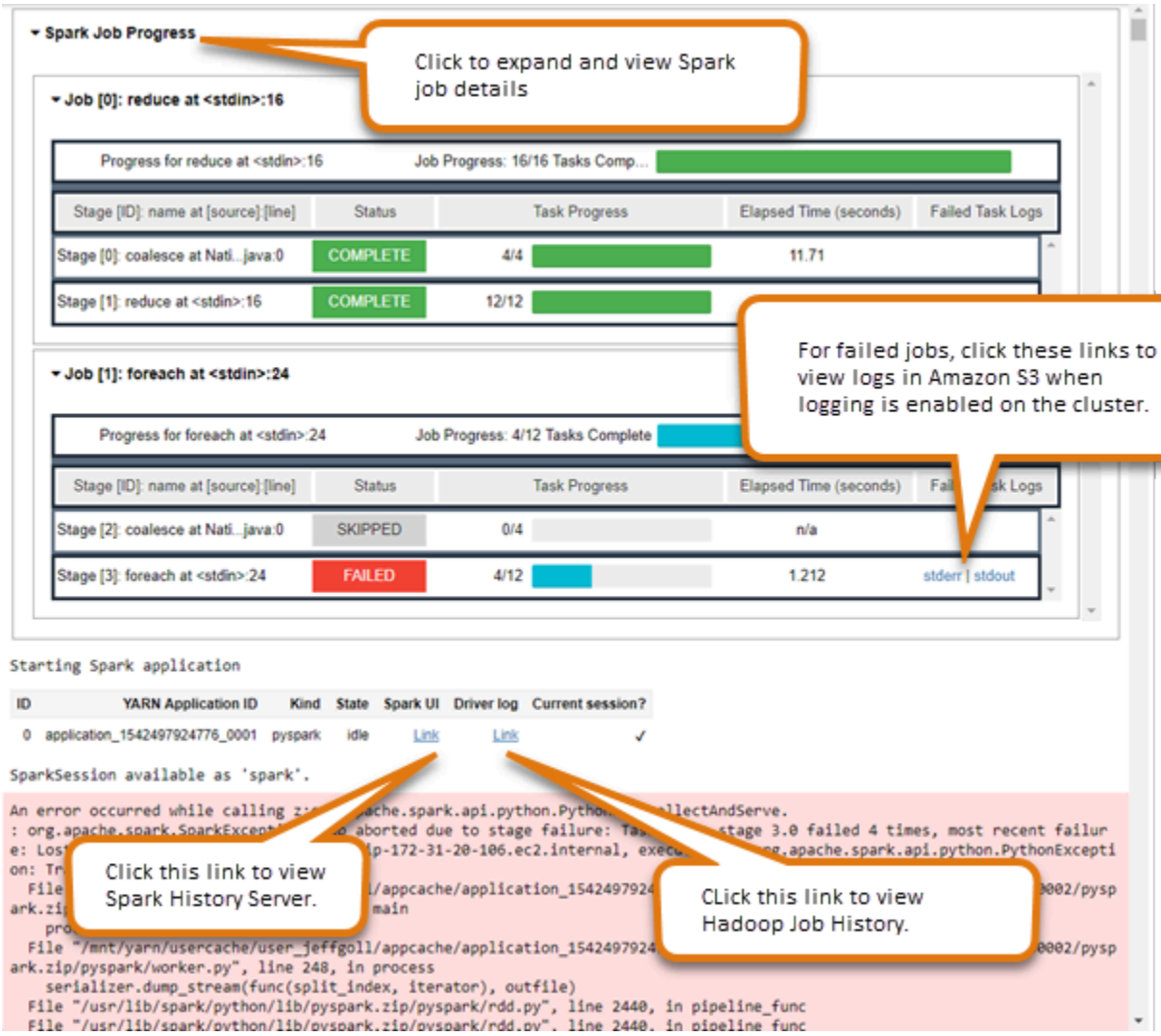
```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
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

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 primaire 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. 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 Afficher des fichiers journaux archivés dans Amazon S3.

Voici un exemple de surveillance d'une tâche Spark.



Sécurité et contrôle d'accès des blocs-notes EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Plusieurs fonctionnalités sont disponibles pour vous aider à adapter la sécurité des 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 fonctionnalités s'ajoutent aux fonctionnalités de sécurité disponibles pour Amazon EMR et les clusters Amazon EMR. Pour plus d'informations, consultez Sécurité dans Amazon EMR.

- Vous pouvez utiliser des déclarations AWS Identity and Access Management de politique ainsi que des balises de bloc-notes pour limiter l'accès. Pour plus d'informations, consultez Fonctionnement d'Amazon EMR avec IAM et Exemple de déclarations de stratégie basées sur l'identité pour les bloc-notes EMR.
- Les groupes de sécurité Amazon EC2 agissent comme un pare-feu virtuel qui contrôle le trafic réseau entre l'instance primaire 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.
- Vous spécifiez un rôle AWS de service qui détermine les autorisations dont dispose un bloc-notes EMR lorsqu'il interagit avec d'autres AWS services. Pour plus d'informations, consultez Rôle de service pour Blocs-notes EMR.

Installation et utilisation des noyaux et des bibliothèques

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

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 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 de l'accès externe pour différentes configurations réseau, consultez la rubrique Scénarios et exemples dans le Guide de l'utilisateur d'Amazon VPC.

Les applications EMR Serverless sont fournies avec les bibliothèques préinstallées suivantes pour Python et : PySpark

- Bibliothèques Python : ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, scipy
- PySpark bibliothèques —ggplot,matplotlib,numpy,pandas,plotly,bokeh,scikit-learn,scipy, scipy

Installation des noyaux et des bibliothèques Python sur le nœud primaire d'un cluster

Avec Amazon EMR en version 5.30.0 et ultérieure, à l'exception de la version 6.0.0, vous pouvez installer des noyaux et des bibliothèques Python supplémentaires sur le nœud primaire 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 primaire. 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.

> **Note**
>
> Pour les versions 5.30.1, 5.31.0 et 6.1.0 d'Amazon EMR, vous devez prendre des mesures supplémentaires afin d'installer les noyaux et les bibliothèques sur le nœud primaire d'un cluster.
> Pour activer cette fonctionnalité, procédez comme suit :
>
> 1. Assurez-vous que la politique d'autorisations attachée à la fonction du service pour les blocs-notes EMR autorise l'action suivante :
>
>    ```
>    elasticmapreduce:ListSteps
>    ```
>
>   Pour plus d'informations, consultez la rubrique Fonction du service pour les bloc-notes EMR.
> 2. Utilisez le AWS CLI pour exécuter une étape sur le cluster qui configure les Notebooks EMR, comme indiqué dans l'exemple suivant. Vous devez utiliser le nom d'étape `EMRNotebooksSetup`. Remplacez *`us-east-1`* par la région dans laquelle réside votre

cluster. Pour plus d'informations sur l'ajout d'étapes, consultez la rubrique Ajout d'étapes à un cluster à l'aide de la AWS CLI.

```
aws emr add-steps --cluster-id MyClusterID --steps
 Type=CUSTOM_JAR,Name=EMRNotebooksSetup,ActionOnFailure=CONTINUE,Jar=s3://us-
east-1.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
awssupportdatasvcs.com/bootstrap-actions/EMRNotebooksSetup/emr-notebooks-
setup.sh"]
```

Vous pouvez installer des noyaux et des bibliothèques à l'aide de `pip` ou de `conda` dans le répertoire `/emr/notebook-env/bin` du nœud primaire.

Example – Installation de bibliothèques Python

À partir du noyau Python3, exécutez la commande magique `%pip` directement depuis une cellule du bloc-notes pour installer les bibliothèques Python.

```
%pip install pmdarima
```

Vous devrez peut-être redémarrer le noyau pour utiliser les packages mis à jour. Vous pouvez également utiliser la commande magique de Spark `%%sh` pour invoquer `pip`.

```
%%sh
/emr/notebook-env/bin/pip install -U matplotlib
/emr/notebook-env/bin/pip install -U pmdarima
```

Lorsque vous utilisez un PySpark noyau, vous pouvez soit installer des bibliothèques sur le cluster à l'aide de `pip` commandes, soit utiliser des bibliothèques adaptées à un bloc-notes à partir d'un bloc-notes. PySpark

Pour exécuter des commandes `pip` sur le cluster depuis le terminal, connectez-vous d'abord au nœud primaire via SSH, comme le montrent les commandes suivantes.

```
sudo pip3 install -U matplotlib
sudo pip3 install -U pmdarima
```

Vous pouvez également utiliser des bibliothèques adaptées aux blocs-notes. Si vous utilisez des bibliothèques adaptées aux blocs-notes, l'installation de votre bibliothèque est limitée à l'étendue

de votre session et s'effectue sur tous les exécuteurs Spark. Pour plus d'informations, consultez la rubrique relative à l'utilisation de bibliothèques adaptées aux blocs-notes.

Si vous souhaitez empaqueter plusieurs bibliothèques Python dans un PySpark noyau, vous pouvez également créer un environnement virtuel Python isolé. Pour obtenir des exemples d'utilisation, consultez Using Virtualenv.

Pour créer un environnement virtuel Python dans une session, utilisez la propriété Spark `spark.yarn.dist.archives` à partir de la commande magique `%%configure` dans la première cellule du bloc-notes, comme le montre l'exemple suivant.

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
   }
}
```

Vous pouvez également créer un environnement d'exécuteur Spark.

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.executorEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
   }
}
```

Vous pouvez également utiliser `conda` pour installer des bibliothèques Python. Vous n'avez pas besoin d'un accès sudo pour utiliser `conda`. Vous devez vous connecter au nœud primaire à l'aide de SSH, puis exécuter `conda` à partir du terminal. Pour plus d’informations, consultez Connexion au nœud primaire à l'aide de SSH.

Example – Installation de noyaux

L'exemple suivant illustre l'installation du noyau Kotlin à l'aide d'une commande de terminal lorsque vous êtes connecté au nœud primaire d'un cluster :

```
sudo /emr/notebook-env/bin/conda install kotlin-jupyter-kernel -c jetbrains
```

> **Note**
>
> Ces instructions n'installent pas les dépendances du noyau. Si votre noyau comporte des dépendances tierces, vous devrez peut-être effectuer des étapes de configuration supplémentaires pour pouvoir utiliser le noyau avec votre bloc-notes.

Considérations et limites relatives aux bibliothèques adaptées aux blocs-notes

Lorsque vous utilisez des bibliothèques adaptées aux blocs-notes, tenez compte des éléments suivants :

- Les bibliothèques adaptées aux blocs-notes sont disponibles pour les clusters que vous créez avec Amazon EMR versions 5.26.0 et ultérieures.
- Les bibliothèques de type Notebook 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.
- Vous pouvez désinstaller uniquement les bibliothèques qui ont été installées à l'aide de l'API `install_pypi_package`. Vous ne pouvez désinstaller aucune bibliothèque qui a été 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.

Travail avec des bibliothèques adaptées aux blocs-notes

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

Les exemples suivants illustrent des commandes simples permettant de répertorier, d'installer et de désinstaller des bibliothèques depuis une cellule d'un bloc-notes à l'aide PySpark du noyau et des API. Pour des exemples supplémentaires, consultez l'article Installer des bibliothèques Python sur un cluster en cours d'exécution avec EMR Notebooks sur AWS le Big Data Blog.

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 la bibliothèque Celery en tant que bibliothèque limitée aux bloc-notes.

```
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 la bibliothèque Arrow en tant que bibliothèque limitée aux bloc-notes, avec une spécification de la version de la bibliothèque et 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 des blocs-notes EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

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

- AWS CodeCommit
- GitHub
- Bitbucket
- GitLab

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 consulter l'historique de vos modifications et les inverser de manière sélective.
- Collaboration – Les collaborateurs qui travaillent dans différents blocs-notes peuvent partager du code via des référentiels Git distants. 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 blocs-notes Jupyter présentant des techniques d'analyse de données ou d'apprentissage automatique sont disponibles dans des référentiels hébergés publiquement, tels que. GitHub Vous pouvez associer vos blocs-notes à un référentiel pour réutiliser les blocs-notes Jupyter contenus dans ce référentiel.

Pour utiliser vos référentiels Git avec les blocs-notes EMR, ajoutez les référentiels en tant que ressources dans la console Amazon EMR, associez les informations d'identification des référentiels nécessitant une authentification et liez-les à vos blocs-notes. Vous pouvez afficher une liste des référentiels stockés dans votre compte ainsi que les détails concernant chaque référentiel dans la console Amazon EMR. Vous pouvez associer un référentiel Git existant à un bloc-notes lorsque de sa création.

Rubriques

- Prérequis et considérations
- Ajout d'un référentiel Git à Amazon EMR
- Mise à jour ou suppression d'un référentiel Git
- Association ou dissociation d'un référentiel Git
- Création d'un nouveau bloc-notes avec un référentiel Git associé
- Utilisation de référentiels Git dans un bloc-notes

Prérequis et considérations

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Tenez compte des éléments suivants lorsque vous prévoyez d'intégrer un référentiel Git aux blocs-notes EMR.

AWS CodeCommit

Si vous utilisez un CodeCommit dépôt, vous devez utiliser les informations d'identification Git et HTTPS avec CodeCommit. Les clés SSH et le protocole HTTPS avec l'assistant AWS CLI d'identification ne sont pas pris en charge. CodeCommit ne prend pas en charge les jetons d'accès

personnels (PAT). Pour plus d'informations, voir Utilisation d'IAM avec CodeCommit : informations d'identification Git, clés SSH et clés d' AWS accès dans le guide de l'utilisateur IAM et Configuration pour les utilisateurs HTTPS à l'aide des informations d'identification Git dans le guide de l'AWS CodeCommit 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 les blocs-notes EMR et vos groupes de sécurité disposent des paramètres et autorisations appropriés. Vous pouvez également configurer des référentiels Git que vous hébergez sur un réseau privé en suivant les instructions de Configuration d'un référentiel Git hébergé sur un serveur privé pour les blocs-notes EMR.

- Accès à internet du cluster – 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, consultez la rubrique Options Amazon VPC.

  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, consultez la rubrique Spécification des groupes de sécurité EC2 pour les bloc-notes EMR.

  > **Important**
  >
  > Si l'interface réseau est lancée dans un sous-réseau public, elle ne pourra pas communiquer avec l'internet par l'intermédiaire d'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, une politique d'autorisation the section called “Rôle de Blocs-notes EMR” doit être jointe pour autoriser cette `secretsmanager:GetSecretValue` action.

Configuration d'un référentiel Git hébergé sur un serveur privé pour les blocs-notes EMR

Suivez les instructions ci-dessous pour configurer des référentiels hébergés sur un serveur privé pour les blocs-notes EMR. Vous devez fournir un fichier de configuration contenant des informations sur

vos serveurs DNS et Git. Amazon EMR utilise ces informations pour configurer les blocs-notes EMR qui peuvent acheminer le trafic vers vos référentiels hébergés sur un serveur privé.

Prérequis

Avant de configurer un référentiel Git hébergé sur un serveur privé pour les blocs-notes EMR, vous devez disposer des éléments suivants :

- Amazon S3 Control Emplacement où les fichiers de votre bloc-notes EMR seront enregistrés.

Configuration d'un ou de plusieurs référentiels Git hébergés sur un serveur privé pour les blocs-notes EMR

1. Créez un fichier de configuration à l'aide du modèle fourni. Incluez les valeurs suivantes pour chaque serveur Git que vous souhaitez spécifier dans votre configuration :

   - **`DnsServerIpV4`** – L'adresse IPv4 de votre serveur DNS. Si vous fournissez des valeurs à la fois pour `DnsServerIpV4` et `GitServerIpV4List`, la valeur de `DnsServerIpV4` est prioritaire et sera utilisée pour résoudre votre `GitServerDnsName`.

     > **Note**
     >
     > Pour utiliser des référentiels Git hébergés sur un serveur privé, votre serveur DNS doit autoriser l'accès entrant depuis les blocs-notes EMR. Nous vous recommandons vivement de protéger votre serveur DNS contre tout autre accès non autorisé.

   - **`GitServerDnsName`** – Le nom DNS de votre serveur Git. Par exemple `"git.example.com"`.
   - **`GitServerIpV4List`** – Une liste d'adresses IPv4 appartenant à votre ou vos serveurs Git.

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
```

```
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. Enregistrez votre fichier de configuration sous `configuration.json`.
3. Chargez le fichier de configuration dans l'emplacement de stockage Amazon S3 que vous avez désigné dans un dossier appelé `life-cycle-configuration`. Par exemple, si votre emplacement S3 par défaut est `s3://DOC-EXAMPLE-BUCKET/notebooks`, votre fichier de configuration doit se trouver à l'adresse `s3://DOC-EXAMPLE-BUCKET/notebooks/`**`life-cycle-configuration/configuration.json`**.

> **Important**
>
> Nous vous recommandons vivement de limiter l'accès à votre dossier `life-cycle-configuration` aux seuls administrateurs des blocs-notes EMR et à la fonction du service associée aux blocs-notes EMR. Vous devez également protéger `configuration.json` contre tout accès non autorisé. Pour des instructions, consultez la rubrique Contrôle de l'accès à un compartiment avec des politiques utilisateur ou Bonnes pratiques de sécurité pour Amazon S3.

Pour des instructions sur le chargement, consultez les rubriques Création d'un dossier et Chargement d'objets dans le Guide de l'utilisateur d'Amazon Simple Storage Service.

Ajout d'un référentiel Git à Amazon EMR

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Reportez-vous aux sections suivantes pour savoir comment ajouter un référentiel Git à un bloc-notes EMR dans l'ancienne console ou à un espace de travail EMR Studio dans la nouvelle console.

New console

Les blocs-notes EMR étant des espaces de travail EMR Studio dans la nouvelle console, vous pouvez suivre les instructions de Lier des référentiels Git à un Workspace EMR Studio pour associer jusqu'à trois référentiels Git à votre espace de travail.

Vous pouvez également utiliser l'extension JupyterLab Git. Choisissez l'icône Git dans la barre latérale gauche de votre bloc-notes Jupyterlab pour accéder à l'extension. Pour plus d'informations sur l'extension, consultez le dépôt GitHub jupyterlab-git.

Pour associer un référentiel Git à un espace de travail, l'administrateur de Studio doit prendre des mesures pour configurer Studio afin d'autoriser la liaison entre les référentiels Git. Pour plus d'informations, consultez Établissez l'accès et les autorisations pour les référentiels Git.

Old console

Ajout d'un référentiel Git en tant que ressource dans votre compte Amazon EMR à l'aide de l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce.
2. Choisissez Git repositories (Référentiels Git), puis Add repository (Ajouter un référentiel).
3. Pour Nom du référentiel, saisissez 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, entrez l'URL du référentiel. Lorsque vous utilisez un CodeCommit référentiel, il s'agit de l'URL qui est copiée lorsque vous choisissez Cloner l'URL puis Cloner HTTPS, par exemple`https://git-codecommit.us-west-2.amazonaws.com/v1/repos/MyCodeCommitRepoName`.
5. Pour Branch (Branche), entrez un nom de branche.
6. Pour les informations d'identification Git, choisissez les options selon les instructions suivantes. Vous pouvez utiliser un nom d'utilisateur et un mot de passe Git ou un jeton d'accès personnel (PAT) pour vous authentifier dans votre référentiel. Les blocs-notes EMR accèdent à vos informations d'identification Git à l'aide des secrets stockés dans Secrets Manager.

> **Note**
>
> Si vous utilisez un GitHub référentiel, nous vous recommandons d'utiliser un jeton d'accès personnel (PAT) pour vous authentifier. À compter du 13 août 2021, les mots de passe ne GitHub seront plus acceptés lors de l'authentification des opérations Git. Pour plus d'informations, consultez l'article sur les exigences d'authentification par jeton pour les opérations Git dans The GitHub Blog.

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 rattaché à un nom d'utilisateur et un mot de passe Git, le secret doit être au format `{"gitUsername": " MyUserName ", "gitPassword": " MyPassword "}`.

Option	Description
Création d'un secret	Choisissez cette option pour associer des informations d'identification Git existante s à un nouveau secret que vous créez dans 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 référenti el, sélectionnez Nom d'utilisateur et mot de passe, entrez le nom secret à utiliser dans Secrets Manager, puis indiquez le nom d'utilisateur et le mot de passe à rattacher au secret. – OU – Si vous utilisez un jeton d'accès personnel pour accéder au référentiel, sélectionnez Jeton d'accès personnel (PAT), saisissez le nom du secret à utiliser dans Secrets Manager, puis saisissez votre jeton d'accès personnel. Pour plus d'informations, consultez Création d'un jeton d'accès personnel pour la ligne de commande GitHub et de 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. Choisissez Add repository (Ajouter un référentiel).

Mise à jour ou suppression d'un référentiel Git

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Reportez-vous aux sections suivantes pour savoir comment supprimer un référentiel Git d'un bloc-notes EMR dans l'ancienne console, ou d'un espace de travail EMR Studio dans la nouvelle console.

New console

Les blocs-notes EMR étant des espaces de travail EMR Studio dans la nouvelle console, vous pouvez vous reporter à Lier des référentiels Git à un Workspace EMR Studio pour plus d'informations sur le travail avec des référentiels Git dans votre espace de travail. Mais pour le moment, vous ne pouvez pas supprimer les référentiels Git des espaces de travail.

Old console

Mise à jour d'un référentiel Git dans l'ancienne console

1. Sur la page Git repositories (Référentiels Git), choisissez le référentiel à mettre à jour.
2. Dans la page du référentiel, choisissez Edit repository (Modifier le référentiel).
3. Mettez à jour Git credentials (Informations d'identification Git) sur la page du référentiel.

Suppression d'un référentiel Git de l'ancienne console

1. Sur la page Git repositories (Référentiels Git), choisissez le référentiel à supprimer.
2. Sur la page du référentiel, sélectionnez tous les blocs-notes actuellement liés au référentiel. Choisissez Unlink notebook (Annuler le lien du bloc-notes).
3. Sur la page du référentiel, choisissez Delete (Supprimer).

> **Note**
>
> Pour supprimer le référentiel Git local d'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. 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

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Suivez les étapes suivantes pour associer ou dissocier un référentiel Git d'un bloc-notes EMR dans l'ancienne console ou d'un espace de travail EMR Studio dans la nouvelle console.

New console

Les blocs-notes EMR étant des espaces de travail EMR Studio dans la nouvelle console, vous pouvez vous reporter à Lier des référentiels Git à un Workspace EMR Studio pour plus d'informations sur le travail avec des référentiels Git dans votre espace de travail. Mais pour le moment, vous ne pouvez pas supprimer les référentiels Git des espaces de travail.

Old console

Pour associer un référentiel Git à un bloc-notes EMR

Le référentiel peut être lié à un bloc-notes une fois que celui-ci est Ready (Prêt).

1. Dans la liste Notebooks (Bloc-notes) choisissez le bloc-notes que vous souhaitez mettre à jour.

2. Dans la section Git repositories (Référentiels Git) de la page Notebook (Bloc-notes) choisissez Link new repository (Lier un nouveau référentiel).
3. Dans la liste des référentiels de la fenêtre Link Git repository to notebook (Lier un référentiel Git au bloc-notes) sélectionnez un ou plusieurs référentiels que vous souhaitez lier à votre bloc-notes, puis choisissez Link repository (Lier le référentiel).

Ou

1. Sur la page Git repositories (Référentiels Git) choisissez le référentiel que vous souhaitez lier à votre bloc-notes.
2. Dans la liste EMR notebooks (Bloc-notes EMR), choisissez Link new notebook (Lier un nouveau bloc-notes) pour lier ce référentiel à un bloc-notes existant.

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

1. Dans la liste Notebooks (Bloc-notes) choisissez le bloc-notes que vous souhaitez mettre à jour.
2. Dans la liste Git repositories (Référentiels Git), sélectionnez le référentiel pour lequel vous souhaitez annuler le lien à votre bloc-notes, puis choisissez Unlink repository (Annuler le lien du référentiel).

Ou

1. Sur la page Git repositories (Référentiels Git), choisissez le référentiel à mettre à jour.
2. Dans la liste EMR notebooks (Bloc-notes EMR), sélectionnez le bloc-notes pour lequel vous souhaitez annuler le lien au référentiel, puis choisissez Unlink notebook (Annuler le lien du bloc-notes).

> **Note**
>
> La création d'un lien entre un référentiel Git et un bloc-notes clone le référentiel distant sur votre bloc-notes Jupyter local. La dissociation du référentiel Git d'un bloc-notes ne fait que déconnecter le bloc-notes du référentiel à distance, mais ne supprime pas 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.

Statut	Signification
Linking (Liaison en cours)	Le référentiel Git est en train d'être lié au bloc-notes. Lorsque l'état du référentiel est Linking (Liaison en cours), vous ne pouvez pas arrêter le bloc-notes.
Linked (Lié)	Le référentiel Git est lié au bloc-notes. Lorsque le référentiel est à l'état Linked (Lié) il 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. Lorsque le référentiel est à l'état Unlinking (Annulation du lien en cours), vous ne pouvez pas arrêter le bloc-notes. 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.
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

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer

de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Création d'un bloc-notes et association de celui-ci à des référentiels Git dans l'ancienne console Amazon EMR

1. Suivez les instructions décrites dans Création d'un bloc-notes.
2. Pour Security group (Groupe de sécurité), choisissez Use your own security group (Utiliser 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, consultez la rubrique Spécification des groupes de sécurité EC2 pour les bloc-notes EMR.

3. Pour Git repositories (Référentiels Git), choisissez le référentiel à associer au bloc-notes.
   1. Choisissez un référentiel stocké en tant que ressource dans votre compte, puis choisissez Save (Enregistrer).
   2. Pour ajouter un nouveau référentiel en tant que ressource dans votre compte, choisissez add a new repository (ajouter un nouveau référentiel). Exécutez le flux de travail Add repository (Ajouter un référentiel) dans une nouvelle fenêtre.

Utilisation de référentiels Git dans un bloc-notes

> **Note**
>
> Les notebooks EMR sont disponibles sous forme d'espaces de travail EMR Studio dans la console. Le bouton Créer un espace de travail de la console vous permet de créer de nouveaux blocs-notes. Pour accéder aux Workspaces ou en créer, les utilisateurs EMR Notebooks doivent disposer d'autorisations de rôle IAM supplémentaires. Pour plus

d'informations, consultez Amazon EMR Notebooks are Amazon EMR Studio Workspaces dans la console et Amazon EMR.

Vous pouvez choisir d'ouvrir dans JupyterLab ou d'ouvrir dans Jupyter lorsque vous ouvrez un bloc-notes.

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 primary
```

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

Si vous choisissez d'ouvrir le bloc-notes avec une JupyterLab interface, vous pouvez utiliser l'extension JupyterLab Git préinstallée. Pour plus d'informations sur l'extension, consultez jupyterlab-git.

Planification et configuration des 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 :

- La région dans laquelle doit s'exécuter cluster, l'emplacement et le mode de stockage des données et l'affichage des résultats. veuillez consulter Configuration de l'emplacement de cluster et du stockage de données.
- Si vous exécutez des clusters Amazon EMR sur Outposts ou dans des Local Zones. Voir Clusters EMR sur AWS Outposts ou Clusters EMR sur les Zones Locales AWS.
- La nature d'un cluster (longue durée, transitoire) et les logiciels qu'il exécute. Consultez Configuration d'un cluster pour qu'il continue ou se résilie après l'exécution de l'étape et Configuration des logiciels de cluster.
- Si un cluster a un seul nœud primaire ou trois nœuds primaires. veuillez consulter Planification et configuration des nœuds primaires.
- Le matériel et les options réseau qui optimisent le coût, les performances et la disponibilité pour votre application. veuillez consulter Configuration du matériel et de la mise en réseau d'un cluster.
- Le mode de configuration des clusters afin de pouvoir les gérer plus facilement, et surveiller l'activité, les performances et la santé. Consultez Configuration de la journalisation et du débogage du cluster et Clusters de balise.
- Authentification et autorisation de l'accès aux ressources du cluster, et chiffrement des données. veuillez consulter Sécurité dans Amazon EMR.
- Le mode d'intégration à d'autres logiciels et services. veuillez consulter Intégration de pilotes et d'applications tierces.

Lancement rapide d'un cluster

Pour lancer rapidement un cluster à l'aide de la console

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

2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sur la page de Création d'un cluster, entrez ou sélectionnez des valeurs pour les champs fournis. Le panneau récapitulatif permanent affiche une vue en temps réel des options de cluster actuellement sélectionnées. Sélectionnez un titre dans le panneau récapitulatif pour accéder à la section correspondante et apporter des modifications. Le nom de votre cluster ne peut pas contenir les caractères <, >, $, | ou `(backtick). Vous devez effectuer toutes les configurations requises avant de pouvoir choisir Créer un cluster.
4. Choisissez Créer un cluster pour accepter la configuration comme indiqué.
5. La page des détails du cluster s'ouvre. Trouvez le statut du cluster à côté du nom du cluster. Le statut doit passer de Démarrage à Exécution puis à Attente au cours du processus de création du cluster. Il se peut que vous deviez cliquer sur l'icône d'actualisation en haut à droite ou actualiser votre navigateur pour recevoir les mises à jour.

   Lorsque le statut passe à Attente, votre cluster est opérationnel et prêt à accepter des étapes et des connexions SSH.

Configuration de l'emplacement de cluster et du stockage de données

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 chargement 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

- Choisissez une AWS région
- Gestion du stockage et des systèmes de fichiers
- Préparation des données d'entrée
- Configuration d'un emplacement de sortie

Choisissez une AWS région

Amazon Web Services s'exécutent 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 principal), 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 vous êtes inscrit à un compte Compte AWS le 17 mai 2017 ou après cette date, la région par défaut lorsque vous accédez à une ressource AWS Management Console est USA Est (Ohio) (us-east-2) ; pour les anciens comptes, la région par défaut est USA Ouest (Oregon) (us-west-2) ou USA Est (Virginie du Nord) (us-east-1). Pour plus d’informations, consultez Régions et points de terminaison.

Certaines AWS fonctionnalités ne sont disponibles que dans certaines régions. 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 les AWS ressources qui seront utilisées avec le cluster. Le tableau suivant met en correspondance les noms des régions entre les services. Pour obtenir la liste des régions Amazon EMR, consultez Régions AWS et points de terminaison dans le Référence générale d'Amazon Web Services.

Choix d'une région à l'aide de la console

Votre région par défaut est affichée à gauche des informations de votre compte dans la barre de navigation. Pour changer de région dans les nouvelles et les anciennes consoles, choisissez le menu déroulant Région et sélectionnez une nouvelle option.

Spécifiez une région à l'aide du AWS CLI

Spécifiez une région par défaut à l' AWS CLI aide de la aws configure commande ou de la variable d'`AWS_DEFAULT_REGION`environnement. Pour plus d'informations, consultez la section Configuration de la AWS région dans le guide de AWS Command Line Interface l'utilisateur.

Choix d'une région à l'aide d'un kit SDK ou de l'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 SDK AWS , 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.

Gestion du stockage et des systèmes de fichiers

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**
>
> Depuis la version 5.22.0 d'Amazon EMR, Amazon EMR AWS utilise Signature version 4 exclusivement pour authentifier les demandes adressées à Amazon S3. Les versions antérieures d'Amazon EMR utilisent la version 2 de AWS Signature dans certains cas, sauf si les notes de publication indiquent que la version 4 de Signature est utilisée exclusivement. Pour plus d'informations, consultez les sections Authentification des demandes (AWS Signature version 4) et Authentification des demandes (AWS Signature version 2) dans le manuel Amazon Simple Storage Service Developer Guide.

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 pour une utilisation avec Hadoop,

Système de fichiers	Préfixe	Description
		tout en fournissant des fonctionnalités telles que le chiffrement côté serveur Amazon S3, read-after-write la cohérence et la cohérence des listes. **Note** Amazon EMR utilisait auparavant les systèmes de fichiers `s3n` et `s3a`. Bien que les deux fonctionnent, nous vous recommandons d'utiliser le schéma d'URI `s3` pour optimiser les performances, la sécurité et la fiabilité.
système de fichiers local		Le système de fichiers local fait référence à un disque connecté localement. Lorsqu'un cluster Hadoop est créé, chaque nœud est créé à partir d'une instance EC2 qui est associée à un bloc préconfiguré de stockage sur disque préinstallé appelé stockage d'instance. Les données des volumes de stockage d'instance sont conservées uniquement pendant la durée de vie de leur 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 de base de données Amazon EC2. Le système de fichiers local est utilisé par HDFS, mais Python s'exécute également à partir du système de fichiers local et vous pouvez choisir de stocker des fichiers d'application supplémentaires sur des volumes de stockage d'instance.

Système de fichiers	Préfixe	Description
(Hérité) Système de fichiers à blocs Amazon S3	`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.

Accès aux systèmes de fichiers

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 qui ne spécifient pas de préfixe dans l'URI vers le 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. Grâce à la fonctionnalité de téléchargement en plusieurs parties fournie par Amazon EMR via AWS le SDK Java, vous pouvez télécharger des fichiers d'une taille maximale de 5 To vers le système de fichiers natif Amazon S3, et le système de fichiers par 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
```

Préparation des données d'entrée

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 courant consiste à charger les données d'entrée dans Amazon S3. Amazon EMR fournit des outils permettant à votre cluster d'importer ou de lire des données depuis 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 de saisie qu'Amazon EMR peut accepter
- Comment transférer des données dans Amazon EMR

Types de saisie qu'Amazon EMR peut accepter

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 d'un format donné dans HDFS. Pour plus d'informations, consultez https://cwiki.apache.org/confluence/display/Hive/ SerDe.

Comment transférer des données dans 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 DistributedCache de Hadoop pour transférer des fichiers 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

- Chargement de données vers Amazon S3
- Charger des données avec AWS DataSync
- Importation de fichiers à l'aide du cache distribué
- Comment traiter les fichiers compressés
- Importer des données DynamoDB dans Hive
- Connexion aux données avec AWS Direct Connect
- Chargement de grandes quantités de données avec AWS Snowball

Chargement de données vers Amazon S3

Pour plus d'informations sur la manière de charger des objets sur Amazon S3, consultez la section Ajout d'un objet à votre compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service. Pour plus d'informations sur l'utilisation d'Amazon S3 avec Hadoop, consultez http://wiki.apache.org/hadoop/AmazonS3.

Rubriques

- Création et configuration d'un compartiment Amazon S3
- Configuration d'un chargement partitionné pour Amazon S3
- Bonnes pratiques
- Chargement de données vers Amazon S3 Express One Zone

Création et configuration d'un compartiment Amazon S3

Amazon EMR utilise le AWS SDK for Java avec Amazon S3 pour stocker les données d'entrée, les fichiers journaux et les données de sortie. Amazon S3 fait référence à ces emplacements de stockage en tant que 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 de l'utilisateur Amazon Simple Storage Service.

Cette section explique comment utiliser Amazon S3 AWS Management Console pour créer puis 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 :

- Pour créer un compartiment à l'aide de la console, consultez Création d'un compartiment dans le Guide de l'utilisateur Amazon S3.
- Pour créer et utiliser des compartiments à l'aide du AWS CLI, consultez la section Utilisation de commandes S3 de haut niveau AWS Command Line Interface dans le guide de l'utilisateur Amazon S3.
- Pour créer un compartiment à l'aide d'un kit SDK, consultez la section Exemples de création d'un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Pour utiliser des compartiments à l'aide de curl, consultez Outil d'authentification Amazon S3 pour curl.
- Pour plus d'informations sur la spécification de compartiments spécifiques à une région, consultez la section Accès à un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.
- Pour utiliser des compartiments utilisant des points d'accès Amazon S3, consultez la section Utilisation d'un alias de type compartiment pour votre point d'accès dans le Guide de l'utilisateur Amazon S3. Vous pouvez facilement utiliser les points d'accès Amazon S3 avec l'alias du point d'accès Amazon S3 au lieu du nom du compartiment Amazon S3. Vous pouvez utiliser l'alias du point d'accès Amazon S3 pour les applications existantes et nouvelles, notamment Spark, Hive, Presto et d'autres.

> **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.

Configuration d'un chargement partitionné pour Amazon S3

Amazon EMR prend en charge le téléchargement partitionné d'Amazon S3 via le SDK for AWS 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 toutes les parties de l'objet chargées, Amazon S3 les assemble et crée l'objet.

Pour plus d'informations, consultez la section Présentation de chargement partitionné dans le Guide de l'utilisateur Amazon Simple Storage Service.

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 Guide de version Amazon EMR.

Nom de paramètre de configuration	Valeur par défaut	Description
`fs.s3n.multipart.uploads.enabled`	`true`	Type booléen qui indique s'il convient d'activer les chargements partitionnés. Lorsque la vue cohérente d'EMRFS est activée, 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 chargement de partie lorsque le chargement partitionné est activé. La valeur minimale est `5242880` (5 Mo). Si une valeur plus

Nom de paramètre de configuration	Valeur par défaut	Description
		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, veuillez consulter la rubrique
`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.

Nom de paramètre de configuration	Valeur par défaut	Description
`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.

Désactivation des chargements partitionnés

Console

Pour désactiver les téléchargements partitionnés avec la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Paramètres du logiciel, saisissez la configuration suivante : `classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer cluster, choisissez Créer un cluster.

CLI

Pour désactiver le téléchargement partitionné à l'aide du 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`.

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* par le nom de votre paire de clés EC2.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.1.0 --applications Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --configurations file://myConfig.json
```

API

Pour désactiver le chargement partitionné à l'aide de l'API

- Pour plus d'informations sur l'utilisation des chargements partitionnés Amazon S3 par programmation, consultez la section Utilisation du kit SDK AWS pour Java pour le chargement partitionné dans le Guide de l'utilisateur Amazon Simple Storage Service.

  Pour plus d'informations sur le AWS SDK pour Java, AWS consultez la section SDK pour Java.

Bonnes pratiques

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

Activation de la gestion des versions

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 Utilisation de la gestion des versions dans le Guide de l'utilisateur Amazon Simple Storage Service.

Nettoyer les chargements partitionnés échoués

Les composants du cluster EMR utilisent des téléchargements partitionnés via le SDK for AWS Java avec les API Amazon S3 pour écrire des fichiers journaux et générer des données vers Amazon S3 par défaut. Pour en savoir plus sur la modification des propriétés liées à cette configuration qui utilise Amazon EMR, consultez Configuration d'un chargement partitionné pour Amazon S3. Parfois, le chargement d'un fichier volumineux peut se traduire par un chargement 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 :

- Pour les compartiments que vous utilisez avec Amazon EMR, utilisez une règle de configuration dans Amazon S3 pour supprimer les chargements partitionnés incomplets trois jours après la date de début de chargement. Les règles de configuration de cycle de vie vous permettent de contrôler la classe de stockage et la durée de vie de vos objets. Pour plus d'informations, consultez Gestion du cycle de vie de l'objet et Interruption de chargements partitionnés inachevés à l'aide de la stratégie de cycle de vie de compartiment.
- Activez la fonctionnalité de nettoyage partitionné d'Amazon EMR en définissant `fs.s3.multipart.clean.enabled` sur `true` et en réglant d'autres paramètres de nettoyage. Cette fonctionnalité est utile à haut volume, grande échelle et avec des clusters dotés d'une disponibilité limitée. Dans ce cas, le paramètre `DaysAfterIntitiation` d'une règle de configuration du cycle de vie peut être trop long, même s'il est défini à son minimum, ce qui provoque des pics de stockage sur Amazon S3. Le nettoyage en plusieurs parties d'Amazon EMR permet un contrôle plus précis. Pour plus d’informations, consultez Configuration d'un chargement partitionné pour Amazon S3.

Gérer les repères de versions

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 d'un compartiment avec la gestion des versions dans le Guide de l'utilisateur Amazon Simple Storage Service.

Bonnes pratiques en matière de performances

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 Guide de l'utilisateur Amazon Simple Storage Service.

Chargement de données vers Amazon S3 Express One Zone

Présentation

Avec Amazon EMR 6.15.0 et versions ultérieures, vous pouvez utiliser Amazon EMR avec Apache Spark conjointement avec la classe de stockage Amazon S3 Express One Zone pour améliorer les performances de vos tâches Spark. S3 Express One Zone est une classe de stockage S3 destinée aux applications qui accèdent fréquemment aux données avec des centaines de milliers de requêtes par seconde. À son lancement, S3 Express One Zone offre la latence la plus faible et les meilleures performances de stockage d’objets cloud dans Amazon S3.

Prérequis

- Autorisations S3 Express One Zone : lorsque S3 Express One Zone effectue initialement une action telle que `GET`, `LIST` ou `PUT` sur un objet Amazon S3, la classe de stockage appelle `CreateSession` pour vous. Votre politique IAM doit accorder l’autorisation `s3express:CreateSession` pour que le connecteur S3A puisse invoquer l’API `CreateSession`. Pour obtenir un exemple de politique avec cette autorisation, voir la rubrique Premiers pas avec Amazon S3 Express One Zone.
- Connecteur S3A : pour configurer votre cluster Spark pour accéder aux données d’un compartiment Amazon S3 utilisant la classe de stockage S3 Express One Zone, vous devez utiliser le connecteur Apache Hadoop S3A. Pour utiliser le connecteur, assurez-vous que tous les URI S3 utilisent le schéma `s3a`. Si ce n’est pas le cas, vous pouvez modifier l’implémentation du système de fichiers que vous utilisez pour les schémas `s3` et `s3n`.

Pour modifier le schéma `s3`, spécifiez les configurations de cluster suivantes :

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Pour modifier le schéma `s3n`, spécifiez les configurations de cluster suivantes :

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3n.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3n.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Premiers pas avec Amazon S3 Express One Zone

Rubriques

- Création d’une politique d’autorisation
- Création et configuration de votre cluster
- Présentation des configurations

Création d’une politique d’autorisation

Avant de créer un cluster utilisant Amazon S3 Express One Zone, vous devez créer une politique IAM à attacher au profil d’instance Amazon EC2 du cluster. La politique doit disposer des autorisations permettant d’accéder à la classe de stockage S3 Express One Zone. L’exemple de politique suivant montre l’autorisation requise. Après avoir créé la politique, associez-la au rôle de profil d’instance que vous utilisez pour créer votre cluster EMR, comme décrit dans la section Création et configuration de votre cluster.

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:s3express:region-code:account-id:bucket/DOC-EXAMPLE-
BUCKET",
            "Action": [
                "s3express:CreateSession"
            ]
        }
    ]

 }
```

Création et configuration de votre cluster

Créez ensuite un cluster qui exécute Spark avec S3 Express One Zone. La procédure suivante explique de manière générale comment créer un cluster dans la AWS Management Console :

1. Accédez à la console Amazon EMR et sélectionnez Clusters dans la barre latérale. Ensuite, choisissez Créer un cluster.
2. Sélectionnez Amazon EMR `emr-6.15.0` ou une version ultérieure.
3. Sélectionnez l'offre d'applications interactives Spark, puis toutes les autres applications que vous souhaitez inclure dans votre cluster. Vous devez au moins inclure Spark et Hadoop dans votre cluster.
4. Pour activer Amazon S3 Express One Zone, entrez une configuration similaire à l'exemple suivant dans la section Paramètres du logiciel. Les configurations et les valeurs recommandées sont présentées dans la section Présentation des configurations qui suit cette procédure.

   ```
   [
     {
       "Classification": "core-site",
       "Properties": {
         "fs.s3a.aws.credentials.provider":
    "software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider",
         "fs.s3a.change.detection.mode": "none",
         "fs.s3a.endpoint.region": "aa-example-1",
         "fs.s3a.select.enabled": "false"
       }
   ```

```
  },
  {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
    }
  }
]
```

5. Dans la section Profil d'instance EC2 pour Amazon EMR, choisissez d'utiliser un rôle existant et utilisez un rôle associé à la politique que vous avez créée dans la section Création d'une politique d'autorisation ci-dessus.
6. Configurez le reste des paramètres de votre cluster en fonction de votre application, puis sélectionnez Créer un cluster.

Présentation des configurations

Les tableaux suivants présentent les configurations et les valeurs suggérées à spécifier lorsque vous configurez un cluster qui utilise S3 Express One Zone avec Amazon EMR, comme indiqué dans la section Création et configuration de votre cluster.

Configurations S3A

Paramètre	Valeur par défaut	Valeur suggérée	Explication
`fs.s3a.aws.credentials.provider`	Si aucune valeur n'est spécifiée, utilise les éléments de la `AWSCredentialProviderList` dans l'ordre suivant : `TemporaryAWSCredentialsProvider`, `SimpleAWSCredentialsProvider`	`software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider`	Le rôle de profil d'instance Amazon EMR doit être associé à la politique qui autorise le système de fichiers S3A à appeler `s3express:CreateSession`. Les autres fournisseurs d'identifiants fonctionnent également s'ils disposent des

Paramètre	Valeur par défaut	Valeur suggérée	Explication
	`r` , `EnvironmentVariableCredentialsProvider` , `IAMInstanceCredentialsProvider` .		autorisations S3 Express One Zone.
`fs.s3a.endpoint.region`	null	L' Région AWS endroit où vous avez créé le bucket.	La logique de résolution de région ne fonctionne pas avec la classe de stockage S3 Express One Zone.
`fs.s3a.select.enabled`	`true`	`false`	L'option `select` d'Amazon S3 n'est pas prise en charge avec la classe de stockage S3 Express One Zone.
`fs.s3a.change.detection.mode`	`server`	none	Détection des modifications par des tâches S3A reposant sur la vérification des `etags` basées sur MD5. La classe de stockage S3 Express One Zone ne prend pas en charge les `checksums` MD5.

Configurations Spark

Paramètre	Valeur par défaut	Valeur suggérée	Explication
`spark.sql.sources.fastS3PartitionDiscovery.enabled`	`true`	`false`	L'optimisation interne utilise un paramètre d'API S3 que la classe de stockage S3 Express One Zone ne prend pas en charge.

Considérations

Tenez compte des points suivants lorsque vous intégrez Apache Spark sur Amazon EMR à la classe de stockage S3 Express One Zone :

- La classe de stockage Amazon S3 Express One Zone est prise en charge avec Amazon EMR 6.15.0 et versions ultérieures.
- Le connecteur S3A est nécessaire pour utiliser la classe de stockage S3 Express One Zone avec Amazon EMR. Seul le connecteur S3A dispose des fonctionnalités et des classes de stockage nécessaires pour interagir avec la classe de stockage S3 Express One Zone. Pour savoir comment configurer le connecteur, voir la rubrique the section called "Prérequis".
- La classe de stockage Amazon S3 Express One Zone n'est prise en charge avec Spark que sur un cluster Amazon EMR qui s'exécute sur Amazon EC2.
- La classe de stockage Amazon S3 Express One Zone prend uniquement en charge le chiffrement SSE-S3. Pour plus d'informations, voir la rubrique Chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3).
- La classe de stockage Amazon S3 Express One Zone ne prend pas en charge les écritures avec le `FileOutputCommitter` S3A. Les écritures avec le `FileOutputCommitter` S3A sur des compartiments S3 Express One Zone entraînent l'erreur suivante : InvalidStorageClass: The storage class you specified is not valid.
- La classe de stockage Amazon S3 Express One Zone n'est prise en charge ni avec Amazon EMR sans serveur ni Amazon EMR sur EKS.

Charger des données avec AWS DataSync

AWS DataSync est un service de transfert de données en ligne qui simplifie, automatise et accélère le processus de transfert des données entre votre stockage sur site et AWS les services de stockage ou entre les services AWS de stockage. DataSync prend en charge divers systèmes de stockage sur site tels que le système de fichiers distribué Hadoop (HDFS), les serveurs de fichiers NAS et le stockage d'objets autogéré.

La façon la plus courante d'obtenir des données sur un cluster est de charger les données sur Amazon S3 et d'utiliser les fonctionnalités intégrées d'Amazon EMR pour charger les données sur votre cluster.

DataSync peut vous aider à accomplir les tâches suivantes :

- Répliquer HDFS sur votre cluster Hadoop vers Amazon S3 pour assurer la continuité des activités
- Copier HDFS sur Amazon S3 pour remplir vos lacs de données
- Transférer des données entre le HDFS de votre cluster Hadoop et Amazon S3 à des fins d'analyse et de traitement

Pour télécharger des données dans votre compartiment S3, vous devez d'abord déployer un ou plusieurs DataSync agents sur le même réseau que votre espace de stockage sur site. Un agent est une machine virtuelle (VM) utilisée pour lire ou écrire des données depuis un emplacement autogéré. Vous activez ensuite vos agents dans le compartiment S3 Compte AWS et à l' Région AWS endroit où se trouve celui-ci.

Une fois votre agent activé, vous créez un emplacement source pour votre stockage sur site, un emplacement de destination pour votre compartiment S3 et une tâche. Une tâche est un ensemble de deux emplacements (source et destination) et un ensemble d'options par défaut que vous utilisez pour contrôler le comportement de la tâche.

Enfin, vous exécutez votre DataSync tâche pour transférer les données de la source vers la destination.

Pour plus d'informations, consultez Getting started with AWS DataSync.

Importation de fichiers à l'aide du cache distribué

Rubriques

- Types de fichier pris en charge

- Emplacement des fichiers mis en cache
- Accès aux fichiers mis en cache à partir d'applications de streaming
- Accès aux fichiers mis en cache à partir d'applications de streaming

DistributedCache 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 DistributedCache 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, rendez-vous sur http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/filecache/ DistributedCache .html.

Vous invoquez DistributedCache 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 Création d'actions d'amorçage pour installer des logiciels supplémentaires.

Types de fichier pris en charge

DistributedCache 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 correspondent à un ou plusieurs fichiers packagés à l'aide d'un utilitaire, comme `gzip`. DistributedCachetransmet les fichiers compressés à chaque nœud principal et décompresse l'archive lors de la mise en cache. DistributedCache prend en charge les formats de compression suivants :

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

Emplacement des fichiers mis en cache

DistributedCache copie les fichiers uniquement vers les nœuds principaux. Si le cluster ne comprend aucun nœud principal, DistributedCache copie les fichiers sur le nœud primaire.

DistributedCache 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`, indique 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 de l'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, DistributedCache le place dans le répertoire `archive`. Si vous mettez en cache une archive, DistributedCache 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 DistributedCache uniquement lorsque vous utilisez le streaming.

Accès aux fichiers mis en cache à partir d'applications de streaming

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.

Accès aux fichiers mis en cache à partir d'applications de streaming

Vous pouvez utiliser le AWS Management Console et AWS CLI pour créer des clusters utilisant le cache distribué.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour spécifier les fichiers de cache distribué avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Étapes, choisissez Ajouter une étape. Cela ouvre la boîte de dialogue Ajouter une étape. Dans le champ Arguments, incluez les fichiers et les archives à enregistrer dans le cache. 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 voulez ajouter un fichier individuel au cache distribué, spécifiez `-cacheFile` suivi du nom et de l'emplacement du fichier, du signe dièse (#) et du nom que vous voulez donner au fichier lorsqu'il est placé dans le cache local. L'exemple suivant montre comment ajouter un fichier individuel au cache distribué.

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file-name#cache-file-name
   ```

   Si vous voulez 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. L'exemple suivant montre comment ajouter un fichier d'archive au cache distribué.

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/archive-name#cache-archive-name
   ```

   Entrez les valeurs appropriées dans les autres champs de la boîte de dialogue. Les options diffèrent selon le type d'étape. Pour ajouter votre étape et quitter la boîte de dialogue, choisissez Ajouter une étape.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour spécifier des fichiers de cache distribués avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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 choisissez 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 voulez ajouter un fichier individuel au cache distribué, spécifiez `-cacheFile` suivi du nom et de l'emplacement du fichier, du signe dièse (#) et du nom que vous voulez donner au fichier lorsqu'il est placé dans le cache local. L'exemple suivant montre comment ajouter un fichier individuel au cache distribué.

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file_name#cache_file_name
   ```

   Si vous voulez 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. L'exemple suivant montre comment ajouter un fichier d'archive au cache distribué.

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/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.

CLI

Pour spécifier des fichiers de cache distribués à l'aide du AWS CLI

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

  Si vous voulez ajouter un fichier individuel au cache distribué, spécifiez `-cacheFile` suivi du nom et de l'emplacement du fichier, du signe dièse (#) et du nom que vous voulez donner au fichier lorsqu'il est placé dans le cache local.

  Si vous voulez 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. L'exemple suivant montre comment ajouter un fichier d'archive au cache distribué.

  Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le 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 primaire 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 primaire 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`.

Comment traiter les fichiers compressés

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.

Importer des données DynamoDB dans 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 plus d'informations, consultez Exportation, importation, interrogation et jointure de tables dans DynamoDB à l'aide d'Amazon EMR.

Connexion aux données avec AWS Direct Connect

AWS Direct Connect est un service que vous pouvez utiliser pour établir une connexion réseau dédiée privée à Amazon Web Services depuis votre centre de données, votre bureau ou votre environnement de colocation. Si vous disposez de grandes quantités de données d'entrée, leur utilisation AWS Direct Connect peut réduire les coûts de votre réseau, augmenter le débit de bande passante et fournir une expérience réseau plus cohérente que les connexions Internet. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Direct Connect.

Chargement de grandes quantités de données avec AWS Snowball

AWS Snowball est un service que vous pouvez utiliser pour transférer rapidement de grandes quantités de données entre Amazon Simple Storage Service (Amazon S3) et votre site de stockage de données sur site. faster-than-internet Snowball prend en charge deux types de tâches : les tâches d'importation et les tâches d'exportation. La fonction d'importation implique le transfert de données d'une source sur site vers un compartiment Amazon S3. La fonction d'exportation implique le transfert de données d'un compartiment Amazon S3 vers une source sur site. Pour les deux types de tâches, les appareils Snowball sécurisent et protègent vos données pendant que les transporteurs régionaux les transportent entre Amazon S3 et votre site de stockage de données sur site. Les appareils Snowball sont physiquement robustes et protégés par le AWS Key Management Service ()AWS KMS. Pour plus d'informations, consultez le Manuel du développeur AWS Snowball Edge.

Configuration d'un emplacement de sortie

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 plus d’informations, consultez les rubriques suivantes :

Rubriques

- Création et configuration d'un compartiment Amazon S3
- Quels formats peuvent être renvoyés par Amazon EMR ?

- Comment écrire des données dans un compartiment Amazon S3, qui ne vous appartient pas
- Compression de la sortie de votre cluster

Création et configuration d'un compartiment Amazon S3

Amazon EMR (Amazon EMR) utilise Amazon S3 pour stocker les données d'entrée, les fichiers journaux et les données de sortie. Amazon S3 fait référence à ces emplacements de stockage en tant que 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.

Pour créer un compartiment Amazon S3, suivez les instructions de la page Création d'un compartiment du Guide du développeur Amazon Simple Storage Service.

> **Note**
>
> Si vous activez la journalisation dans l'assistant de création d'un compartiment, cela active uniquement les journaux d'accès au compartiment, et non les journaux de cluster.

> **Note**
>
> Pour plus d'informations sur la spécification de compartiments spécifiques à une région, consultez la section Compartiments et régions du manuel Amazon Simple Storage Service Developer Guide et les points de terminaison régionaux disponibles pour les SDK. AWS

Après avoir créé votre compartiment, vous pouvez définir les autorisations appropriées sur celui-ci. Généralement, vous (le propriétaire) vous accordez un accès en lecture et en écriture. Nous vous recommandons vivement de suivre les Bonnes pratiques de sécurité pour Amazon S3 lors de la configuration de votre compartiment.

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`

Quels formats peuvent être renvoyés par Amazon EMR ?

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 depuis 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 utilisez un cluster et que vous souhaitez que la sortie soit écrite dans le compartiment Amazon S3 d'un autre AWS utilisateur, et que vous souhaitez que cet autre AWS utilisateur puisse lire cette sortie, vous devez faire deux choses :

- Demandez à l'autre AWS utilisateur de vous accorder des autorisations d'écriture pour son compartiment Amazon S3. Le cluster que vous lancez fonctionne avec vos AWS informations d'identification, de sorte que tous les clusters que vous lancez pourront également écrire dans le compartiment de cet autre AWS utilisateur.
- Définissez des autorisations de lecture pour l'autre AWS utilisateur sur les fichiers que vous ou le cluster 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 manière dont AWS l'autre utilisateur peut vous accorder l'autorisation d'écrire des fichiers dans le compartiment Amazon S3 de l'autre utilisateur, consultez la section Modification des autorisations du compartiment dans le guide de l'utilisateur d'Amazon Simple Storage Service.

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 compartiment reçoit l'autorisation `Permission.FullControl` . Le propriétaire du compartiment n'est pas nécessairement le propriétaire de l'objet.
`BucketOwnerRead`	Spécifie que le propriétaire du compartiment reçoit l'autorisation `Permission.Read` . Le propriétaire du compartiment n'est pas nécessairement le propriétaire de l'objet.

Liste ACL prête à l'emploi	Description
`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 primaire à l'aide de SSH.

  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 primaire à l'aide de SSH.

  L'exemple suivant définit l'option de `fs.s3.canned.acl` configuration sur BucketOwnerFullControl, ce qui donne au propriétaire du compartiment Amazon S3 le contrôle total du 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
  ```

Compression de la sortie de votre cluster

Rubriques

- Compression de données de sortie
- Compression de données intermédiaire
- Utilisation de la bibliothèque Snappy avec Amazon EMR

Compression de données de sortie

Cela compresse les données de sortie de votre tâche Hadoop. Si vous utilisez TextOutputFormat le résultat est un fichier texte gzippé. Si vous SequenceFiles écrivez dessus, le résultat est compressé en interne. SequenceFile 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);
```

Compression de données intermédiaire

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);
```

Utilisation de la bibliothèque Snappy avec 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 primaires

Lorsque vous lancez un cluster Amazon EMR, vous pouvez choisir d'avoir un ou trois nœuds primaires dans votre cluster. La haute disponibilité des flottes d'instances est prise en charge par les versions 5.36.1, 5.36.2, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0 et supérieures d'Amazon EMR. Pour les groupes d’instances, la haute disponibilité est prise en charge avec les versions 5.23.0 et les versions ultérieures d’Amazon EMR. Amazon EMR peut tirer parti des groupes de placement Amazon EC2 pour garantir que les nœuds primaires sont placés sur du matériel sous-jacent distinct afin d’améliorer la disponibilité des clusters. Pour plus d’informations, consultez Intégration d'Amazon EMR aux groupes de placement EC2.

Un cluster Amazon EMR doté de plusieurs nœuds primaires offre les principaux avantages suivants :

- Le nœud primaire n'est plus un point de défaillance unique. Si l'un des nœuds primaires échoue, le cluster utilise les deux autres nœuds primaires et s'exécute sans interruption. En attendant, Amazon EMR remplace automatiquement le nœud primaire en échec par un nouveau qui est configuré avec les mêmes configurations et actions d'amorçage.

- Amazon EMR active les fonctionnalités de haute disponibilité Hadoop de HDFS NameNode et YARN ResourceManager et prend en charge la haute disponibilité pour quelques autres applications open source.

  Pour plus d'informations sur la façon dont un cluster Amazon EMR avec plusieurs nœuds primaires prend en charge les applications open source et d'autres fonctions Amazon EMR, consultez Applications et fonctionnalités prises en charge.

> **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 Amazon EMR avec plusieurs nœuds primaires 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
- Lancer un cluster Amazon EMR doté de plusieurs nœuds primaires
- Intégration d'Amazon EMR aux groupes de placement EC2
- Considérations et bonnes pratiques

Applications et fonctionnalités prises en charge

Cette rubrique fournit des informations sur les fonctionnalités de haute disponibilité Hadoop de HDFS NameNode et YARN dans un cluster ResourceManager Amazon EMR, ainsi que sur la manière dont les fonctionnalités de haute disponibilité fonctionnent avec les applications open source et les autres fonctionnalités d'Amazon EMR.

Haute disponibilité de HDFS

Un cluster Amazon EMR doté de plusieurs nœuds primaires donne accès à la fonctionnalité de haute disponibilité HDFS NameNode dans Hadoop. Pour plus d'informations, consultez Haute disponibilité de HDFS.

Dans un cluster Amazon EMR, au moins deux nœuds distincts sont configurés en tant que. NameNodes L'un NameNode est dans un `active` État et les autres dans un `standby` État. En cas de `active` NameNode défaillance du nœud, Amazon EMR lance un processus de basculement HDFS automatique. Un nœud `standby` NameNode devient `active` et prend en charge toutes les opérations du client dans le cluster. Amazon EMR remplace le nœud en échec par un nouveau, puis le rejoint en tant que `standby`.

> **Note**
>
> Dans les versions 5.23.0 et 5.30.1 incluses d'Amazon EMR, seuls deux des trois nœuds principaux exécutent HDFS. NameNode

Si vous avez besoin de savoir lequel NameNode est`active`, vous pouvez utiliser SSH pour vous connecter à n'importe quel nœud principal du cluster et exécuter la commande suivante :

```
hdfs haadmin -getAllServiceState
```

La sortie répertorie les nœuds sur lesquels NameNode il est installé et leur état. Par exemple,

```
ip-##-#-#-##1.ec2.internal:8020 active
ip-##-#-#-##2.ec2.internal:8020 standby
ip-##-#-#-##3.ec2.internal:8020 standby
```

FIL À HAUTE DISPONIBILITÉ ResourceManager

Un cluster Amazon EMR avec plusieurs nœuds principaux active la fonctionnalité de ResourceManager haute disponibilité YARN dans Hadoop. Pour plus d'informations, consultez la section ResourceManager Haute disponibilité.

Dans un cluster Amazon EMR comportant plusieurs nœuds principaux, YARN ResourceManager s'exécute sur les trois nœuds principaux. L'un ResourceManager est en `active` état et les deux autres sont en `standby` état. En cas de `active` ResourceManager défaillance du nœud principal, Amazon EMR lance un processus de basculement automatique. Un nœud principal doté d'un `standby` ResourceManager prend en charge toutes les opérations. Amazon EMR remplace le nœud principal défaillant par un nouveau, qui rejoint ensuite le ResourceManager quorum en tant que. `standby`

Vous pouvez vous connecter à « http://*master-public-dns-name*:8088/cluster » pour n'importe quel nœud primaire, qui vous dirige automatiquement vers le gestionnaire de ressources `active`. Pour déterminer quel gestionnaire de ressources est `active`, utilisez SSH pour vous connecter à un nœud primaire dans le cluster. Ensuite, exécutez la commande suivante pour obtenir la liste des trois nœuds primaires et leur statut :

```
yarn rmadmin -getAllServiceState
```

Applications prises en charge dans un cluster Amazon EMR avec plusieurs nœuds primaires

Vous pouvez installer et exécuter les applications suivantes sur un cluster Amazon EMR comportant plusieurs nœuds primaires. Pour chaque application, le processus de basculement du nœud primaire (ou nœud primaire) varie.

Application	Disponibilité lors du basculement du nœud primaire	Remarques
Flink	Disponibilité non affectée par le basculement du nœud primaire	Les tâches Flink sur Amazon EMR s'exécutent en tant qu'applications YARN. Flink JobManagers fonctionne en tant que YARN ApplicationMasters sur les nœuds principaux. Le JobManager n'est pas affecté par le processus de basculement du nœud principal. Si vous utilisez Amazon EMR version 5.27.0 ou antérieure, il s' JobManager agit d'un point de défaillance unique. En cas d' JobManager échec, il perd tous les états des tâches et ne reprend pas les tâches en cours d'exécution. Vous pouvez activer la JobManager haute disponibilité en configurant le nombre de tentatives d'application, le point de contrôle et en activant ZooKeeper le stockage d'état pour Flink. Pour plus d'informations, consultez

Application	Disponibilité lors du basculement du nœud primaire	Remarques
		Configuration de Flink sur un cluster Amazon EMR doté de plusieurs nœuds primaires. À partir de la version 5.28.0 d'Amazon EMR, aucune configuration manuelle n'est nécessaire pour activer la haute disponibilité. JobManager
Ganglia	Disponibilité non affectée par le basculement du nœud primaire	Ganglia est disponible sur tous les nœuds primaires. Ainsi, Ganglia peut continuer à s'exécuter pendant le processus de basculement du nœud primaire.
Hadoop	Haute disponibilité	HDFS NameNode et YARN ResourceManager basculent automatiquement vers le nœud de secours lorsque le nœud principal actif tombe en panne.
HBase	Haute disponibilité	HBase bascule automatiquement vers le nœud en veille lorsque le nœud primaire actif échoue. Si vous vous connectez à HBase via un serveur Thrift ou REST, vous devez passer à un autre nœud primaire lorsque le nœud primaire actif échoue.
HCatalog	Disponibilité non affectée par le basculement du nœud primaire	HCatalog est construit sur le métastore Hive, qui existe en dehors du cluster. HCatalog reste disponible pendant le processus de basculement du nœud primaire.

Application	Disponibilité lors du basculement du nœud primaire	Remarques
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 primaire. 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 primaires . Lorsque le nœud primaire actif échoue, vous perdez l'accès à la session Livy actuelle et vous devez créer une nouvelle session Livy sur un autre nœud primaire ou sur le nouveau nœud de remplacement.
Mahout	Disponibilité non affectée par le basculement du nœud primaire	Étant donné que Mahout n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud primaire.
MXNet	Disponibilité non affectée par le basculement du nœud primaire	Étant donné que MXNet n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud primaire.
Phoenix	Haute disponibilité	Phoenix' ne QueryServer fonctionne que sur l'un des trois nœuds principaux. Sur les trois maîtres, Phoenix est configuré pour connecter le 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`

Application	Disponibilité lors du basculement du nœud primaire	Remarques
Pig	Disponibilité non affectée par le basculement du nœud primaire	Étant donné que Pig n'a pas de démon, il n'est pas affecté par le processus de basculement du nœud primaire.
Spark	Haute disponibilité	Toutes les applications Spark s’exécutent dans des conteneurs YARN et peuvent réagir au basculement du nœud primaire 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 primaire.
TensorFlow	Disponibilité non affectée par le basculement du nœud primaire	Comme il n' TensorFlow a pas de daemon, il n'est pas affecté par le processus de basculement du nœud principal.

Application	Disponibilité lors du basculement du nœud primaire	Remarques
Zeppelin	Haute disponibilité	Zeppelin est installé sur les trois nœuds primaires. 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. 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 fonction de basculement automatique HDFS. ZooKeeper fournit un service hautement disponible pour gérer les données de coordination, informer les clients des modifications apportées à ces données et surveiller les clients en cas de défaillance. Pour de plus amples informations, veuillez consulter Basculement automatique de HDFS.

Pour exécuter les applications suivantes dans un cluster Amazon EMR doté de plusieurs nœuds primaires, 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 primaire. Pour les applications suivantes, les composants de service se rétabliront automatiquement pendant le processus de basculement du nœud primaire, mais les tâches actives peuvent échouer et doivent être relancées.

Application	Disponibilité lors du basculement du nœud primaire	Remarques
Hive	Haute disponibilité pour les composants de service uniquement	Un metastore externe pour Hive est requis. Il doit s'agir d'un métastore externe MySQL, car PostgreSQL n'est pas pris en charge pour les clusters multi-maîtres. 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. Oozie-server et oozie-client sont installés sur les trois nœuds primaires. Les clients oozie- sont configurés pour se connecter au serveur oozie-server correct par défaut.
PrestoDB ou PrestoSQL/ Trino	Haute disponibilité pour les composants de service uniquement	Un métastore Hive externe pour PrestoDB (PrestoSQL sur Amazon EMR 6.1.0-6.3.0 ou Trino sur Amazon EMR 6.4.0 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. Presto CLI est installé sur les trois nœuds primaires afin que vous puissiez l'utiliser

Application	Disponibilité lors du basculement du nœud primaire	Remarques
		pour accéder au coordinateur Presto depuis n'importe lequel des nœuds primaires. Presto Coordinator est installé sur un seul nœud primaire. Vous pouvez trouver le nom DNS du nœud primaire sur lequel le coordinateur Presto est installé en appelant l'API Amazon EMR `describe-cluster` et en lisant la valeur renvoyée du champ `MasterPublicDnsNam e` dans la réponse.

> **Note**
>
> Lorsqu'un nœud primaire échoue, votre Java Database Connectivity (JDBC) ou Open Database Connectivity (ODBC) met fin à sa connexion au nœud primaire. 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 primaire en échec.

Comment les fonctionnalités Amazon EMR fonctionnent dans un cluster avec plusieurs nœuds primaires

Connexion aux nœuds primaires à l'aide de SSH

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

En cas de défaillance d'un nœud primaire, votre connexion SSH à ce nœud primaire prend fin. Pour continuer votre travail, vous pouvez vous connecter à l'un des deux autres nœuds primaires. Vous pouvez également accéder au nouveau nœud primaire une fois qu'Amazon EMR remplace celui en échec par un nouveau.

> **Note**
>
> L'adresse IP privée pour le nœud primaire de remplacement reste la même que la précédente. L'adresse IP publique pour le nœud primaire 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' AWS CLI.
> NameNode ne fonctionne que sur deux des nœuds principaux. 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 Amazon EMR avec plusieurs nœuds primaires

Vous pouvez soumettre des étapes à un cluster Amazon EMR avec plusieurs nœuds primaires de la même manière que vous travaillez avec des étapes dans un cluster avec un seul nœud primaire. Pour plus d'informations, consultez Soumission de travail à un cluster.

Voici les points à prendre en compte lors de l'utilisation des étapes dans un cluster Amazon EMR comportant plusieurs nœuds primaires :

- Si un nœud primaire échoue, les étapes en cours d'exécution sur le nœud primaire 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 primaire échoue, l'étape peut continuer et réussir grâce au basculement automatique du nœud primaire.
- Il est recommandé de vérifier le statut des étapes en faisant référence à la sortie des tâches. Par exemple, les MapReduce tâches utilisent un `_SUCCESS` fichier pour déterminer si elles se terminent correctement.
- Il est recommandé de définir le ActionOnFailure paramètre sur CONTINUE, ou CANCEL_AND_WAIT, au lieu de TERMINATE_JOB_FLOW ou TERMINATE_CLUSTER.

Protection contre la résiliation automatique

Amazon EMR active automatiquement la protection contre les résiliations pour tous les clusters comportant plusieurs nœuds primaires et remplace tous les paramètres d'exécution des étapes que vous fournissez lors de la création du cluster. Vous pouvez désactiver la protection contre la résiliation après le lancement du cluster. veuillez consulter Configuration de la protection contre la

résiliation pour les clusters en cours d'exécution. Pour résilier un cluster comportant plusieurs nœuds primaires, vous devez d'abord modifier les attributs du cluster afin de désactiver la protection contre la résiliation. Pour obtenir des instructions, veuillez consulter Résiliation d'un cluster Amazon EMR avec plusieurs nœuds primaires.

Pour plus d'informations sur la protection contre les résiliations, consultez Utilisation de la protection contre la résiliation.

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

Les fonctionnalités Amazon EMR suivantes ne sont actuellement pas disponibles dans un cluster Amazon EMR comportant plusieurs nœuds primaires :

- Blocs-notes EMR
- Accès en un clic au serveur d'historique Spark permanent
- Interfaces utilisateur d'application persistante
- L'accès en un clic aux interfaces utilisateur persistantes des applications n'est actuellement pas disponible pour les clusters Amazon EMR dotés de plusieurs nœuds principaux ou pour les clusters Amazon EMR intégrés à Lake Formation. AWS

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

Lancer un cluster Amazon EMR doté de plusieurs nœuds primaires

Cette rubrique fournit des détails de configuration et des exemples pour le lancement d'un cluster Amazon EMR avec plusieurs nœuds primaires.

> **Note**
>
> Amazon EMR active automatiquement la protection contre la résiliation pour tous les clusters dotés de plusieurs nœuds primaires et remplace tous les paramètres d’arrêt automatique

que vous spécifiez lors de la création du cluster. Pour résilier un cluster comportant plusieurs nœuds primaires, vous devez d'abord modifier les attributs du cluster afin de désactiver la protection contre la résiliation. Pour obtenir des instructions, veuillez consulter Résiliation d'un cluster Amazon EMR avec plusieurs nœuds primaires.

Prérequis

- Vous pouvez lancer un cluster Amazon EMR avec plusieurs nœuds primaires dans des sous-réseaux VPC publics et privés. EC2-Classic n'est pas pris en charge. Pour lancer un cluster Amazon EMR avec plusieurs nœuds primaires 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 l'ID de votre 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 Amazon EMR comportant plusieurs nœuds primaires, vous devez créer un métastore externe. Pour plus d'informations, consultez Configuration d'une métastore 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 plus d'informations, consultez Configuration de Kerberos sur Amazon EMR.

Lancer un cluster Amazon EMR doté de plusieurs nœuds primaires

Vous pouvez lancer un cluster avec plusieurs nœuds primaires lorsque vous utilisez des groupes ou des flottes d'instances. Lorsque vous utilisez des groupes d'instances dotés de plusieurs nœuds primaires, vous devez définir le nombre d'instances sur `3` pour le groupe d'instances du nœud primaire. Lorsque vous utilisez des flottes d'instances dotées de plusieurs nœuds primaires, vous devez définir `TargetOnDemandCapacity` sur `3` et `TargetSpotCapacity` sur `0` pour la flotte d'instances principales, et définir `WeightedCapacity` sur `1` pour chaque type d'instance que vous configurez pour la flotte principale.

Les exemples suivants montrent comment lancer le cluster à l'aide de l'AMI par défaut ou d'une AMI personnalisée avec des groupes et des flottes d'instances :

> **Note**
>
> Vous devez spécifier l'ID de sous-réseau lorsque vous lancez un cluster Amazon EMR avec plusieurs nœuds primaires à l'aide de l' AWS CLI. Remplacez *22XXXX01* et *22XXXX02* par votre identifiant de sous-réseau dans les exemples suivants.

Default AMI, instance groups

Example – Lancement d'un cluster de groupe d'instances Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une AMI par défaut

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--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
```

Default AMI, instance fleets

Example – Lancement d'un cluster de flotte d'instances Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une AMI par défaut

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
```

```
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
```

```
        "Name": "Core - 2"
    }
]' \
 --ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

Custom AMI, instance groups

Example – Lancement d'un cluster de groupe d'instances Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une AMI personnalisée

```
aws emr create-cluster \
--name "custom-ami-ha-cluster" \
--release-label emr-6.15.0 \
--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
```

Custom AMI, instance fleets

Example – Lancement d'un cluster de flotte d'instances Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une AMI personnalisée

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
```

```
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
```

```
        ],
        "Name": "Core - 2"
    }
]' \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID
```

Résiliation d'un cluster Amazon EMR avec plusieurs nœuds primaires

Pour résilier un cluster Amazon EMR avec plusieurs nœuds primaires, vous devez désactiver la protection contre la résiliation avant de résilier le cluster, comme le montre l'exemple suivant. Remplacez *j-3KVTXXXXXX7UG* par votre ID de cluster.

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

Intégration d'Amazon EMR aux groupes de placement EC2

Lorsque vous lancez un cluster à plusieurs nœuds primaires Amazon EMR sur Amazon EC2, vous avez la possibilité d'utiliser des stratégies de groupe de placement pour spécifier la manière dont vous souhaitez déployer les instances de nœuds principaux afin de les protéger contre les pannes matérielles.

Les stratégies de groupes de placement sont prises en charge à partir de la version 5.23.0 d'Amazon EMR en tant qu'option pour plusieurs clusters de nœuds primaires. Actuellement, seuls les types de nœuds primaires sont pris en charge par la stratégie de groupe de placement, et la stratégie `SPREAD` est appliquée à ces nœuds primaires. La stratégie `SPREAD` place un petit groupe d'instances sur un matériel sous-jacent distinct afin de se prémunir contre la perte de plusieurs nœuds primaires en cas de panne matérielle. Notez qu'une demande de lancement d'instance peut échouer si le matériel unique est insuffisant pour répondre à la demande. Pour plus d'informations sur les stratégies de placement et les limites d'EC2, consultez la section Groupes de placement du Guide de l'utilisateur EC2 pour les instances Linux.

Amazon EC2 impose une limite initiale de 500 clusters basés sur la stratégie des groupes de placement qui peuvent être lancés par région. AWS Contactez l' AWS assistance pour demander une

augmentation du nombre de groupes de placement autorisés. Vous pouvez identifier les groupes de placement EC2 créés par Amazon EMR en suivant la paire clé-valeur qu'Amazon EMR associe à la stratégie des groupes de placement Amazon EMR. Pour plus d'informations sur les balises d'instance de cluster EC2, consultez Afficher les instances de cluster dans Amazon EC2.

Associer la politique gérée par le groupe de placement au rôle Amazon EMR

La stratégie de groupe de placement nécessite une politique gérée appelée `AmazonElasticMapReducePlacementGroupPolicy`, qui permet à Amazon EMR de créer, de supprimer et de décrire des groupes de placement sur Amazon EC2. Vous devez associer `AmazonElasticMapReducePlacementGroupPolicy` à la fonction du service pour Amazon EMR avant de lancer un cluster Amazon EMR. doté de plusieurs nœuds primaires.

Vous pouvez également associer la politique gérée `AmazonEMRServicePolicy_v2` au rôle de service Amazon EMR au lieu de la politique gérée par le groupe de placement. `AmazonEMRServicePolicy_v2` permet le même accès aux groupes de placement sur Amazon EC2 que la `AmazonElasticMapReducePlacementGroupPolicy`. Pour plus d'informations, consultez Rôle de service pour Amazon EMR (rôle EMR).

La politique gérée `AmazonElasticMapReducePlacementGroupPolicy` est le texte JSON suivant créé et administré par Amazon EMR.

> **Note**
>
> Étant donné que la stratégie `AmazonElasticMapReducePlacementGroupPolicy` gérée est mise à jour automatiquement, il se peut que la politique affichée ici l'est out-of-date. Utilisez la console AWS de gestion pour consulter la politique actuelle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"*",
         "Effect":"Allow",
         "Action":[
            "ec2:DeletePlacementGroup",
            "ec2:DescribePlacementGroups"
         ]
      },
```

```
      {
         "Resource":"arn:aws:ec2:*:*:placement-group/pg-*",
         "Effect":"Allow",
         "Action":[
            "ec2:CreatePlacementGroup"
         ]
      }
   ]
}
```

Lancement d'un cluster Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une stratégie de groupe de placement

Pour lancer un cluster Amazon EMR doté de plusieurs nœuds primaires à l'aide d'une stratégie de groupe de placement, associez la politique gérée du groupe de placement `AmazonElasticMapReducePlacementGroupPolicy` au rôle Amazon EMR. Pour plus d'informations, consultez Associer la politique gérée par le groupe de placement au rôle Amazon EMR.

Chaque fois que vous utilisez ce rôle pour démarrer un cluster Amazon EMR doré de plusieurs nœuds primaires, Amazon EMR tente de lancer un cluster avec une stratégie `SPREAD` appliquée à ses nœuds primaires. Si vous utilisez un rôle auquel la politique gérée du groupe de placement `AmazonElasticMapReducePlacementGroupPolicy` n'est pas associée, Amazon EMR tente de lancer un cluster Amazon EMR doté de plusieurs nœuds primaires sans stratégie de groupe de placement.

Si vous lancez un cluster Amazon EMR doté de plusieurs nœuds primaires avec le paramètre `placement-group-configs` à l'aide de l'API Amazon EMR ou de la CLI, Amazon EMR lance le cluster uniquement si la politique gérée du groupe de placement `AmazonElasticMapReducePlacementGroupPolicy` est associée au rôle Amazon EMR. Si le rôle Amazon EMR n'est pas associé à la politique, le démarrage du cluster Amazon EMR doté de plusieurs nœuds primaires échoue.

Amazon EMR API

Example Exemple : utilisation d'une stratégie de groupe de placement pour lancer un cluster de groupes d'instances doté de plusieurs nœuds primaires à partir de l'API Amazon EMR

Lorsque vous utilisez l' RunJobFlow action pour créer un cluster Amazon EMR avec plusieurs nœuds principaux, définissez la `PlacementGroupConfigs` propriété sur ce qui suit.

Actuellement, le rôle d'instance `MASTER` utilise automatiquement `SPREAD` comme stratégie de groupe de placement.

```
{
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER"
      }
   ],
   "ReleaseLabel": emr-6.15.0,
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
            "InstanceCount":3,
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- Remplacez *`ha-cluster`* par le nom de votre cluster à haute disponibilité.
- Remplacez *`subnet-22XXXX01`* par votre ID de sous-réseau.
- Remplacez *`ec2_key_pair_name`* par le nom de votre paire de clés EC2 pour ce cluster. La paire de clés EC2 est facultative et obligatoire uniquement si vous souhaitez utiliser SSH pour accéder à votre cluster.

AWS CLI

Example Exemple : utilisation d'une stratégie de groupe de placement pour lancer un cluster de flotte d'instances doté de plusieurs nœuds primaires à partir de la AWS Command Line Interface

Lorsque vous utilisez l' RunJobFlow action pour créer un cluster Amazon EMR avec plusieurs nœuds principaux, définissez la `PlacementGroupConfigs` propriété sur ce qui suit. Actuellement, le rôle d'instance `MASTER` utilise automatiquement `SPREAD` comme stratégie de groupe de placement.

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
```

```
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Core - 2"
    }
]' \
--ec2-attributes '{
    "KeyName": "ec2_key_pair_name",
    "InstanceProfile": "EMR_EC2_DefaultRole",
    "SubnetIds": [
        "subnet-22XXXX01",
        "subnet-22XXXX02"
    ]
}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- Remplacez *ha-cluster* par le nom de votre cluster à haute disponibilité.

- Remplacez *ec2_key_pair_name* par le nom de votre paire de clés EC2 pour ce cluster. La paire de clés EC2 est facultative et obligatoire uniquement si vous souhaitez utiliser SSH pour accéder à votre cluster.
- Remplacez *subnet-22XXXX01* et *subnet-22XXXX02* par vos identifiants de sous-réseau.

Lancer un cluster avec plusieurs nœuds primaires sans stratégie de groupe de placement

Pour qu'un cluster comportant plusieurs nœuds primaires puisse lancer des nœuds primaires sans la stratégie du groupe de placement, vous devez effectuer l'une des opérations suivantes :

- Supprimez la politique gérée du groupe de placement `AmazonElasticMapReducePlacementGroupPolicy` du rôle Amazon EMR, ou
- Lancez un cluster avec plusieurs nœuds primaires avec le paramètre `placement-group-configs` à l'aide de l'API Amazon EMR ou de la CLI en choisissant `NONE` comme stratégie de groupe de placement.

Amazon EMR API

Example – Lancement d'un cluster avec plusieurs nœuds primaires sans stratégie de groupe de placement à l'aide d'Amazon EMRAPI.

Lorsque vous utilisez l' RunJobFlow action pour créer un cluster avec plusieurs nœuds principaux, définissez la `PlacementGroupConfigs` propriété sur ce qui suit.

```
{
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER",
         "PlacementStrategy":"NONE"
      }
   ],
   "ReleaseLabel":"emr-5.30.1",
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
```

```
            "InstanceCount":3,
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- Remplacez *ha-cluster* par le nom de votre cluster à haute disponibilité.
- Remplacez *subnet-22XXXX01* par votre ID de sous-réseau.
- Remplacez *ec2_key_pair_name* par le nom de votre paire de clés EC2 pour ce cluster. La paire de clés EC2 est facultative et obligatoire uniquement si vous souhaitez utiliser SSH pour accéder à votre cluster.

Amazon EMR CLI

Example – Lancement d'un cluster avec plusieurs nœuds primaires sans stratégie de groupe de placement à l'aide de l'Amazon EMRCLI.

Lorsque vous utilisez l' RunJobFlow action pour créer un cluster avec plusieurs nœuds principaux, définissez la `PlacementGroupConfigs` propriété sur ce qui suit.

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER,PlacementStrategy=NONE \
--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
```

- Remplacez *ha-cluster* par le nom de votre cluster à haute disponibilité.
- Remplacez *subnet-22XXXX01* par votre ID de sous-réseau.
- Remplacez *ec2_key_pair_name* par le nom de votre paire de clés EC2 pour ce cluster. La paire de clés EC2 est facultative et obligatoire uniquement si vous souhaitez utiliser SSH pour accéder à votre cluster.

Vérification de la configuration de la stratégie de groupe de placement attachée au cluster avec plusieurs nœuds primaires

Vous pouvez utiliser l'API de description du cluster Amazon EMR pour voir la configuration de la stratégie de groupe de placement attachée au cluster comportant plusieurs nœuds primaires.

Example

```
aws emr describe-cluster --cluster-id "j-xxxxx"
{
   "Cluster":{
      "Id":"j-xxxxx",
      ...
      ...
      "PlacementGroups":[
         {
            "InstanceRole":"MASTER",
            "PlacementStrategy":"SPREAD"
         }
      ]
   }
}
```

Considérations et bonnes pratiques

Tenez compte des éléments suivants lorsque vous créez un cluster Amazon EMR doté de plusieurs nœuds primaires :

> **Important**
>
> Pour lancer des clusters EMR à haute disponibilité dotés de plusieurs nœuds primaires, nous vous recommandons vivement d'utiliser la dernière version d'Amazon EMR. Vous vous

assurez ainsi de bénéficier du plus haut niveau de résilience et de stabilité pour vos clusters à haute disponibilité.

- La haute disponibilité des flottes d'instances est prise en charge par les versions 5.36.1, 5.36.2, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0 et supérieures d'Amazon EMR. Pour les groupes d'instances, la haute disponibilité est prise en charge avec les versions 5.23.0 et les versions ultérieures d'Amazon EMR. Pour plus d'informations, voir À propos des versions d'Amazon EMR.
- Sur les clusters à haute disponibilité, Amazon EMR prend uniquement en charge le lancement de nœuds primaires avec des instances à la demande. Cela garantit une disponibilité maximale pour votre cluster.
- Vous pouvez toujours spécifier plusieurs types d'instances pour la flotte principale, mais tous les nœuds primaires des clusters à haute disponibilité sont lancés avec le même type d'instance, y compris les remplacements des nœuds primaires défectueux.
- Pour qu'un cluster à haute disponibilité doté de plusieurs nœuds primaires continue à fonctionner, deux nœuds primaires sur trois doivent être sains. Par conséquent, si les deux nœuds primaires présentent une défaillance simultanément, votre cluster EMR présentera lui aussi une défaillance.
- Tous les clusters EMR, y compris les clusters à haute disponibilité, sont lancés dans une seule zone de disponibilité. Par conséquent, ils ne peuvent pas tolérer les défaillances des zones de disponibilité. Dans le cas d'une panne d'une zone de disponibilité, vous perdez l'accès au cluster.
- Amazon EMR ne garantit pas la haute disponibilité des applications open source autres que celles qui sont spécifiées dans Applications prises en charge dans un cluster Amazon EMR avec plusieurs nœuds primaires.
- Dans les versions 5.23.0 à 5.30.1 d'Amazon EMR, seuls deux des trois nœuds primaires d'un cluster de groupe d'instances exécutent HDFS NameNode.

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

- Un cluster Amazon EMR comportant plusieurs nœuds primaires ne peut résider que dans une seule zone de disponibilité ou sous-réseau. Amazon EMR ne peut pas remplacer un nœud primaire 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 :

- Nous vous recommandons de lancer au moins quatre nœuds principaux pour vous assurer que les nœuds principaux sont également hautement disponibles. Si vous décidez de lancer un cluster plus petit avec trois nœuds principaux ou moins, définissez `dfs.replication parameter` sur au moins 2 pour que HDFS ait une réplication DFS suffisante. Pour de plus amples informations, veuillez consulter Configuration de HDFS.

> ⚠ Warning
>
> 1. Paramétrer `dfs.replication` sur la valeur 1 avec les clusters de moins de quatre nœuds peut entraîner une perte de données HDFS en cas de panne d'un seul nœud. Nous vous recommandons d'utiliser un cluster comportant au moins quatre nœuds principaux pour les charges de travail de production.
> 2. Amazon EMR n'autorisera pas les clusters à mettre à l'échelle les nœuds principaux situés en dessous de `dfs.replication`. Par exemple, si `dfs.replication = 2`, le nombre minimum de nœuds principaux est 2.
> 3. Lorsque vous utilisez la mise à l'échelle gérée, autoscaling, ou que vous choisissez de redimensionner manuellement votre cluster, nous vous recommandons de définir `dfs.replication` sur une valeur supérieure ou égale à 2.

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

- Amazon EMR ne fournit pas les métriques spécifiques à une application donnée sur HDFS ou YARN. Nous vous reccomandons de configurer des alarmes pour surveiller le nombre d'instances du nœud primaire. Configurez les alarmes à l'aide CloudWatch des métriques Amazon suivantes : `MultiMasterInstanceGroupNodesRunningMultiMasterInstanceGroupNodesRunningPercent` ou`MultiMasterInstanceGroupNodesRequested`. CloudWatch vous informera en cas de défaillance ou de remplacement du nœud principal.
  - Si le `MultiMasterInstanceGroupNodesRunningPercentage` est inférieur à 1,0 et supérieur à 0,5, le cluster peut avoir perdu un nœud primaire. Dans cette situation, Amazon EMR tente de remplacer un nœud primaire.
  - Si le `MultiMasterInstanceGroupNodesRunningPercentage` passe en dessous de 0,5, deux nœuds primaires peuvent avoir échoué. Dans ce cas, le quorum est perdu et le cluster ne peut pas être récupéré. Vous devez migrer manuellement les données hors 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 5.28.0, vous pouvez créer et exécuter des clusters EMR sur. AWS Outposts AWS Outposts permet AWS des services, une infrastructure et des modèles d'exploitation natifs dans les installations sur site. Dans AWS Outposts les environnements, vous pouvez utiliser les mêmes AWS API, outils et infrastructures que ceux que vous utilisez dans le AWS cloud. Amazon EMR on AWS Outposts est idéal pour les charges de travail à faible latence qui doivent être exécutées à proximité des données et des applications sur site. Pour plus d'informations AWS Outposts, consultez le Guide de AWS Outposts l'utilisateur.

Prérequis

Voici les prérequis à satisfaire pour utiliser Amazon EMR sur AWS Outposts :

- Vous devez l'avoir installé et configuré AWS Outposts dans votre centre de données sur site.
- Vous devez disposer d'une connexion réseau fiable entre votre environnement Outpost et une AWS région.
- Vous devez disposer d'une capacité suffisante pour les types d'instances compatibles avec Amazon EMR disponibles 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.
- Lorsque vous utilisez les versions 5.28 AWS Outposts à 6.x d'Amazon EMR, vous ne pouvez utiliser que des compartiments S3 qui stockent des objets dans un format que vous spécifiez. Région AWS Avec Amazon EMR 7.0.0 et versions ultérieures, Amazon EMR activé AWS Outposts est également pris en charge avec le préfixe du S3A client de système de fichiers. `s3a://`
- Seuls les types d'instance suivants sont pris en charge par Amazon EMR sur AWS Outposts :

Classe d'instance	Types d’instances
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 |c5.18xlarge | c5d.xlarge |c5d.2xlarge | c5d.4xlarge | 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 avant-poste et sa AWS région est perdue, vos clusters continueront de fonctionner. Toutefois, vous ne pouvez pas créer de nouveaux clusters ni effectuer de nouvelles actions sur les clusters existants tant que la connectivité n'a pas été rétablie. En cas de défaillance d'instance, l'instance ne sera pas automatiquement remplacée. En outre, les actions telles que l'ajout d'étapes à un cluster en cours d'exécution, la vérification de l'état d'exécution des étapes et l'envoi de CloudWatch métriques et d'événements seront retardées.
- Nous vous recommandons de fournir une connectivité réseau fiable et hautement disponible entre votre avant-poste et la AWS région. Si la connectivité réseau entre votre avant-poste et sa AWS région est perdue pendant plus de quelques heures, les clusters qui ont activé la protection de résiliation continueront de fonctionner, et les clusters qui ont désactivé la protection de résiliation peuvent 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 régionale est utilisée pour un cluster Amazon EMR avec plusieurs nœuds primaires.

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 un sous-réseau Amazon EC2 associé à votre Outpost.

Un Amazon VPC peut couvrir toutes les zones de disponibilité d'une AWS région. AWS Outposts sont des extensions de zones de disponibilité, et vous pouvez étendre un Amazon VPC dans un compte pour couvrir plusieurs zones de disponibilité et les emplacements d'avant-poste associés. Lorsque vous configurez votre Outpost, vous lui associez un groupe de sous-réseaux pour étendre votre environnement VPC régional à votre installation sur site. Les instances Outpost et les services connexes apparaissent comme faisant partie de votre VPC régional, tout comme une zone de disponibilité et ses sous-réseaux associés. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Outposts.

Console

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

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Pour créer un cluster sur AWS Outposts la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.

3. Sous Configuration du cluster, sélectionnez Groupes d'instances ou Parcs d'instances. Choisissez ensuite un type d'instance dans le menu déroulant Choisir le type d'instance EC2 ou sélectionnez Actions, puis et choisissez Ajouter des volumes EBS. Amazon EMR on AWS Outposts prend en charge un nombre limité de volumes et de types d'instances Amazon EBS.
4. Sous Mise en réseau, sélectionnez un sous-réseau EC2 avec un ID d'avant-poste au format suivant : op-123456789.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

Old console

Pour créer un cluster sur AWS Outposts l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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. Dans Configuration matérielle, pour le sous-réseau EC2, sélectionnez un sous-réseau Amazon EC2 avec un ID d'avant-poste au format suivant : op-123456789.
6. Choisissez un type d'instance ou ajoutez des volumes de stockage Amazon EBS pour des groupes d'instances ou des parcs d'instances uniformes. Des types limités de volumes et d'instances Amazon EBS sont pris en charge pour Amazon EMR sur AWS Outposts.

CLI

Pour créer un cluster à l' AWS Outposts aide du AWS CLI

- Pour créer un nouveau cluster Amazon EMR AWS Outposts avec le AWS CLI, spécifiez un sous-réseau EC2 associé à votre Outpost, comme dans l'exemple suivant. Remplacez le *sous-réseau 22XXXX01* par votre propre ID de sous-réseau Amazon EC2.

```
aws emr create-cluster \
```

```
--name "Outpost cluster" \
--release-label emr-7.1.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey SubnetId=subnet-22XXXX01 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Clusters EMR sur les Zones Locales AWS

À partir de la version 5.28.0 d'Amazon EMR, vous pouvez créer et exécuter des clusters Amazon EMR sur un sous-réseau de zones AWS locales en tant qu'extension logique d'une région qui prend en charge les zones locales. AWS Une zone locale permet aux fonctionnalités d'Amazon EMR et à un sous-ensemble de AWS services, tels que les services de calcul et de stockage, d'être situés plus près des utilisateurs afin de fournir un accès à très faible latence aux applications exécutées localement. Pour obtenir la liste des zones locales disponibles, consultez AWS Local Zones. Pour plus d'informations sur l'accès aux zones AWS locales disponibles, voir 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 sur Local Zones. La disponibilité du type d'instance peut varier selon la région.

Classe d'instance	Types d’instances
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 sur les Zones Locales

Créez un cluster Amazon EMR sur les zones AWS locales en lançant 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 les notebooks Amazon EMR ni les connexions directes à Amazon EMR via l'interface VPC endpoint ().AWS PrivateLink

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour créer un cluster sur une zone locale à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Mise en réseau, sélectionnez un sous-réseau EC2 avec un ID de zone local au format suivant : sous-réseau 123abc | us-west-2-lax-1a.
4. Choisissez un type d'instance ou ajoutez des volumes de stockage Amazon EBS pour des groupes d'instances ou des parcs d'instances uniformes.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour créer un cluster sur une zone locale avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.

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 Configuration matérielle, pour Sous-réseau EC2, sélectionnez un sous-réseau EC2 avec un ID de Local Zone 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.

CLI

Pour créer un cluster sur une zone locale avec AWS CLI

- Utilisez la commande create-cluster, ainsi que la commande SubnetId pour la zone locale, comme indiqué dans l'exemple suivant. Remplacez le sous-réseau 22xxxx1234567 par la zone SubnetId locale et remplacez les autres options si nécessaire. Pour plus d'informations, consultez 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, ce qui permet au YARN de NodeManager lancer des conteneurs soit directement sur le cluster Amazon EMR, soit dans 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 YARN NodeManager et des 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.

Les clusters Amazon EMR 6.x sont configurés par défaut pour permettre aux applications YARN, telles que Spark, de 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, nous vous recommandons d'utiliser Amazon ECR, qui est un registre de conteneurs 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 des 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.

Sous-réseau privé

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

Pour plus d'informations sur la manière d' AWS PrivateLink autoriser l'accès à Amazon ECR dans un scénario de sous-réseau privé, consultez Configuration pour AWS PrivateLink 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 local (private) et centos. Pour utiliser d'autres référentiels publics ou Amazon ECR, vous pouvez remplacer les paramètres `docker.trusted.registries` dans `/etc/hadoop/conf/container-executor.cfg` à 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 de 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.

```
[
  {
    "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 cluster Amazon EMR 6.0.0 avec cette configuration à l'aide de AWS Command Line Interface (AWS CLI), créez un fichier nommé `container-executor.json` avec le contenu de la

configuration JSON ontainer-executor précédente. 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 ne connaissez pas Amazon ECR, suivez les instructions de la section Premiers pas avec Amazon ECR et vérifiez que vous avez accès à Amazon ECR à partir de chaque instance de votre cluster Amazon EMR.

Sur EMR 6.0.0 et les versions antérieures, pour accéder à 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 à partir d'Amazon ECR, utilisez la variable d'environnement 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 auprès d'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, il n'est pas nécessaire de configurer manuellement l'authentification auprès d'Amazon ECR. Si un registre Amazon ECR est détecté dans la clé de classification `container-executor`, la fonctionnalité d'authentification automatique Amazon ECR s'active et YARN gère le processus d'authentification lorsque vous soumettez une tâche Spark avec une image ECR. Vous pouvez vérifier si l'authentification automatique est activée en vérifiant `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` sur yarn-site. L'authentification automatique est activée et le paramètre d'authentification YARN est défini sur `true` si `docker.trusted.registries` contient une URL de registre ECR.

Conditions préalables à l'utilisation de l'authentification automatique auprès d'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 n'importe quelle image

Pour plus d'informations, reportez-vous à la section Configuration d'Amazon ECR.

Comment activer l'authentification automatique

Suivez Configuration des registres Docker pour définir un registre Amazon ECR comme registre de confiance et assurez-vous que le référentiel 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 sécurisé, utilisez la classification de configuration pour définir `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` sur `true`.

Comment désactiver l'authentification automatique

Par défaut, l'authentification automatique est désactivée si aucun registre Amazon ECR n'est détecté dans le registre sécurisé.

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

Comment vérifier si l'authentification automatique est activée sur un 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`.

Contrôle de la mise hors service d'un cluster

Cette section décrit les options qui s'offrent à vous pour arrêter les clusters Amazon EMR. Il couvre la résiliation automatique et la protection contre la résiliation, ainsi que la manière dont elles interagissent avec les autres fonctionnalités d'Amazon EMR.

Vous pouvez arrêter un cluster Amazon EMR de différentes manières :

- Résiliation après l'exécution de la dernière étape : créez un cluster transitoire qui s'arrête une fois toutes les étapes terminées.

- Résiliation automatique (après une période d'inactivité) : créez un cluster avec une politique de résiliation automatique qui s'arrête après une durée d'inactivité spécifiée. Pour plus d'informations, consultez Utilisation d'une politique de résiliation automatique.
- Résiliation manuel : créez un cluster de longue durée qui continue de fonctionner jusqu'à ce que vous le résiliez délibérément. Pour plus d'informations sur la résiliation manuelle d'un cluster, consultez Arrêter un cluster.

Vous pouvez également définir une protection contre les interruptions sur un cluster afin d'éviter d'arrêter les instances EC2 par accident ou par erreur.

Lorsqu'Amazon EMR arrête votre cluster, toutes les instances Amazon EC2 du cluster s'arrêtent. Les données du stockage d'instances et des volumes EBS ne sont plus disponibles et ne sont plus 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.

Rubriques

- Configuration d'un cluster pour qu'il continue ou se résilie après l'exécution de l'étape
- Utilisation d'une politique de résiliation automatique
- Utilisation de la protection contre la résiliation

Configuration d'un cluster pour qu'il continue ou se résilie après l'exécution de l'étape

Cette rubrique explique les différences entre l'utilisation d'un cluster de longue durée et la création d'un cluster transitoire qui s'arrête après l'exécution de la dernière étape. Il explique également comment configurer l'exécution des étapes pour un cluster.

Création d'un cluster à long terme

Par défaut, les clusters que vous créez à l'aide de la console ou du AWS CLI sont de longue durée. Les clusters de longue durée continuent de fonctionner, d'accepter du travail et d'accumuler des frais jusqu'à ce que vous preniez des mesures pour les arrêter.

Un cluster de longue durée est efficace dans les situations suivantes :

- Lorsque vous devez interroger des données de manière interactive ou automatique.

- Lorsque vous devez interagir en permanence avec des applications Big Data hébergées sur le cluster.
- Lorsque vous traitez périodiquement un jeu de données si important ou si fréquent qu'il est inefficace de lancer de nouveaux clusters et de charger les données à chaque fois.

Vous pouvez également définir une protection contre la résiliation sur un cluster de longue durée afin d'éviter d'arrêter les instances EC2 par accident ou par erreur. Pour plus d'informations, consultez Utilisation de la protection contre la résiliation.

> **Note**
>
> Amazon EMR active automatiquement la protection contre les résiliations pour tous les clusters comportant plusieurs nœuds primaires et remplace tous les paramètres d'exécution des étapes que vous fournissez lors de la création du cluster. Vous pouvez désactiver la protection contre la résiliation après le lancement du cluster. veuillez consulter Configuration de la protection contre la résiliation pour les clusters en cours d'exécution. Pour résilier un cluster comportant plusieurs nœuds primaires, vous devez d'abord modifier les attributs du cluster afin de désactiver la protection contre la résiliation. Pour obtenir des instructions, veuillez consulter Résiliation d'un cluster Amazon EMR avec plusieurs nœuds primaires.

Configurer un cluster pour qu'il se résilie après l'exécution de l'étape

Lorsque vous configurez la résiliation après l'exécution des étapes, le cluster démarre, exécute des actions d'amorçage, puis exécute les étapes que vous spécifiez. Dès que la dernière étape est terminée, Amazon EMR résilie les instances Amazon EC2 du cluster. L'exécution par étapes est activée par défaut pour les clusters que vous lancez avec l'API Amazon EMR.

Le fait de résilier après l'exécution d'une étape est efficace pour les clusters qui effectuent une tâche de traitement périodique, telle qu'une exécution quotidienne de traitement de données. L'exécution des étapes vous permet également de vous assurer que vous n'êtes facturé que pour le temps nécessaire au traitement de vos données. Pour plus d'informations sur ces étapes, consultez Soumission de travail à un cluster.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

Console

Pour activer la terminaison après l'exécution d'une étape avec la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Étapes, choisissez Ajouter une étape. Dans la boîte de dialogue Ajouter une étape, saisissez les valeurs de champ appropriées. Les options diffèrent selon le type d'étape. Pour ajouter votre étape et quitter la boîte de dialogue, choisissez Ajouter une étape.
4. Sous Résiliation du cluster, cochez la case Résilier le cluster une fois la dernière étape terminée.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

AWS CLI

Pour activer la terminaison après l'exécution d'une étape à l'aide du AWS CLI

- Spécifiez le paramètre `--auto-terminate` quand vous utilisez la commande `create-cluster` pour créer un cluster transitoire.

  L'exemple suivant montre comment utiliser le paramètre `--auto-terminate`. Vous pouvez taper la commande suivante et remplacer *myKey* par le nom de votre paire de clés EC2.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.1.0 \
--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
```

API

Pour désactiver la terminaison après l'exécution d'une étape avec l'API Amazon EMR lors du lancement du cluster

1. Lorsque vous utilisez l'action RunJobFlow pour créer un cluster, définissez la propriété KeepJobFlowAliveWhenNoSteps sur`false`.
2. Pour modifier votre configuration de résiliation après exécution des étapes avec l'API Amazon EMR après le lancement du cluster :

   Utilisez l' SetKeepJobFlowAliveWhenNoSteps action.

Utilisation d'une politique de résiliation automatique

Une politique de résiliation automatique vous permet d'orchestrer le nettoyage des clusters sans avoir à surveiller et à résilier manuellement les clusters inutilisés. Lorsque vous ajoutez une politique de résiliation automatique à un cluster, vous spécifiez la durée d'inactivité après laquelle le cluster doit se résilier automatiquement..

Selon la version publiée, Amazon EMR utilise différents critères pour marquer un cluster comme inactif. Le tableau suivant explique comment Amazon EMR détermine l'inactivité du cluster.

Lorsque vous utilisez...	Un cluster est considéré comme inactif lorsque...
Amazon EMR versions 5.34.0 et ultérieures, et 6.4.0 et versions ultérieures	• Aucune application YARN n'est active • L'utilisation du HDFS est inférieure à 10 % • Aucune connexion à un bloc-notes EMR ou à EMR Studio n'est active • Aucune interface utilisateur d'application intégrée au cluster n'est utilisée • Il n'y a aucune étape en attente
Versions d'Amazon EMR 5.30.0 à 5.33.0 et 6.1.0 à 6.3.0	• Aucune application YARN n'est active • Le cluster n'a aucune tâche Spark active **Note** Amazon EMR marque un cluster comme inactif et peut le résilier automatiquement même si vous avez un noyau Python3 actif. Cela est dû au fait que l'exécution d'un noyau Python3 ne soumet pas de tâche Spark sur le cluster. Pour utiliser l'arrêt automatiq ue avec un noyau Python3, nous vous recommandons d'utiliser Amazon EMR version 6.4.0 ou ultérieure.

> **Note**
>
> Les versions 6.4.0 et ultérieures d'Amazon EMR prennent en charge un fichier sur le cluster pour détecter l'activité sur le nœud primaire : `/emr/metricscollector/isbusy`. Lorsque vous utilisez un cluster pour exécuter des scripts shell ou des applications autres que YARN, vous pouvez régulièrement le toucher ou le mettre à jour `isbusy` pour indiquer à Amazon EMR que le cluster n'est pas inactif.

Vous pouvez associer une politique de résiliation automatique lorsque vous créez un cluster ou lorsque vous ajoutez une politique à un cluster existant. Pour modifier ou désactiver la résiliation automatique, vous pouvez mettre à jour ou supprimer la politique.

Considérations

Tenez compte des fonctions et des limites suivantes avant d'utiliser une politique de résiliation automatique :

- Dans ce qui suit Régions AWS, la résiliation automatique d'Amazon EMR est disponible avec Amazon EMR 6.14.0 et versions ultérieures :
  - Asie-Pacifique (Hyderabad) (ap-south-2)
  - Asie-Pacifique (Jakarta) (ap-southeast-3)
  - Europe (Espagne) (eu-south-2)
- Dans ce qui suit Régions AWS, la résiliation automatique d'Amazon EMR est disponible avec Amazon EMR 5.30.0, 6.1.0 et versions ultérieures :
  - USA Est (Virginie du Nord) (us-east-1)
  - USA Est (Ohio) (us-east-2)
  - USA Ouest (Oregon) (us-west-2)
  - US Ouest (N. California) (us-west-1)
  - Afrique (Le Cap) (af-south-1)
  - Asie-Pacifique (Hong Kong) (ap-east-1)
  - Asie-Pacifique (Mumbai) (ap-south-1)
  - Asie-Pacifique (Séoul) (ap-northeast-2)
  - Asie-Pacifique (Singapour) (ap-southeast-1)
  - Asie-Pacifique (Sydney) (ap-southeast-2)

  - Asie-Pacifique (Tokyo) (ap-northeast-1)
  - Canada (Centre) (ca-central-1)
  - Amérique du Sud (São Paulo) (sa-east-1)
  - Europe (Francfort) (eu-central-1)
  - Europe (Irlande) (eu-west-1)
  - Europe (Londres) (eu-west-2)
  - Europe (Milan) (eu-south-1)
  - Europe (Paris) (eu-west-3)
  - Europe (Stockholm) (eu-north-1)
  - Chine (Beijing) cn-north-1
  - Chine (Ningxia) cn-northwest-1
  - AWS GovCloud (Etats-Unis-Est) (us-gov-east-1)
  - AWS GovCloud (US-Ouest) (us-gov-west-1)
- Le délai d'inactivité est par défaut de 60 minutes (une heure) lorsque vous ne spécifiez pas de montant. Vous pouvez spécifier un délai d'inactivité minimal d'une minute et un délai d'inactivité maximal de 7 jours.
- Avec les versions 6.4.0 et ultérieures d'Amazon EMR, la résiliation automatique est activée par défaut lorsque vous créez un nouveau cluster avec la console Amazon EMR.
- Amazon EMR publie des Amazon CloudWatch métriques haute résolution lorsque vous activez la résiliation automatique d'un cluster. Vous pouvez utiliser ces indicateurs pour suivre l'activité et l'inactivité du cluster. Pour plus d’informations, consultez Métriques de capacité de cluster.
- La résiliation automatique n'est pas prise en charge lorsque vous utilisez des applications non basées sur YARN telles que Presto, Trino ou HBase.
- Pour utiliser la résiliation automatique, le processus metrics-collector doit être en mesure de se connecter au point de terminaison de l'API public pour la résiliation automatique dans API Gateway. Si vous utilisez un nom DNS privé avec Amazon Virtual Private Cloud, la terminaison automatique ne fonctionnera pas correctement. Pour garantir le bon fonctionnement de la résiliation automatique, nous vous recommandons de prendre l'une des mesures suivantes :
  - Supprimez le point de terminaison d'un VPC de l'interface de passerelle d'API de votre Amazon VPC.

  - Suivez les instructions de la section Pourquoi est-ce que je reçois une erreur d'accès interdit HTTP 403 lors de la connexion à mes API passerelles depuis un VPC ? pour désactiver le paramètre de nom DNS privé.
  - Lancez votre cluster dans un sous-réseau privé à la place. Pour plus d'informations, consultez la rubrique sur Sous-réseaux privés.
- (Amazon EMR 5.30.0 et versions ultérieures) Si vous supprimez la règle Autoriser tous les accès sortants par défaut sur 0.0.0.0/ pour le groupe de sécurité principal, vous devez ajouter une règle qui autorise la connectivité TCP sortante à votre groupe de sécurité pour l'accès au service sur le port 9443. Votre groupe de sécurité pour l'accès au service doit également autoriser le trafic TCP entrant sur le port 9443 en provenance du groupe de sécurité principal. Pour plus d'informations sur la configuration des groupes de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux privés).

Autorisations d'utilisation de la résiliation automatique

Avant de pouvoir appliquer et gérer les politiques de résiliation automatique pour Amazon EMR, vous devez associer les autorisations répertoriées dans l'exemple de politique d'autorisation IAM suivant aux ressources IAM qui gèrent votre cluster EMR.

```
{
  "Version": "2012-10-17",
  "Statement": {
      "Sid": "AllowAutoTerminationPolicyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:PutAutoTerminationPolicy",
        "elasticmapreduce:GetAutoTerminationPolicy",
        "elasticmapreduce:RemoveAutoTerminationPolicy"
      ],
      "Resource": "<your-resources>"
    }
}
```

Attacher, mettre à jour ou supprimer une politique de résiliation automatique

Cette section contient des instructions pour vous aider à joindre, mettre à jour ou supprimer une politique de résiliation automatique d'un cluster Amazon EMR. Avant de travailler avec des politiques

de résiliation automatique, assurez-vous de disposer des autorisations IAM nécessaires. veuillez consulter Autorisations d'utilisation de la résiliation automatique.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour associer une politique de résiliation automatique lorsque vous créez un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Résiliation du cluster, sélectionnez Résilier le cluster après une période d'inactivité.
4. Spécifiez le nombre d'heures et de minutes d'inactivité qui peuvent s'écouler avant que le cluster ne se résilie automatiquement. Le temps d'inactivité par défaut est de 1 heure.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer votre cluster, choisissez Créer le cluster.

Pour associer, mettre à jour ou supprimer une politique de résiliation automatique sur un cluster en cours d'exécution avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Propriétés de la page des détails du cluster, recherchez Résiliation du cluster et sélectionnez Modifier.
4. Sélectionnez ou désélectionnez Activer la résiliation automatique pour activer ou désactiver la fonctionnalité. Si vous activez la résiliation automatique, spécifiez le nombre d'heures et de

minutes d'inactivité qui peuvent s'écouler avant que le cluster ne se résilie automatiquement. Sélectionnez ensuite Enregistrer les modifications pour confirmer.

Old console

Pour associer une politique de résiliation automatique lorsque vous créez un cluster avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster.
3. Sous Configuration matérielle, sélectionnez Résiliation automatique.
4. Spécifiez le nombre d'heures et de minutes d'inactivité après lesquelles le cluster doit se résilier automatiquement.. La durée d'inactivité par défaut est d'une heure.
5. Choisissez d'autres paramètres selon les besoins de votre application, puis choisissez Create Cluster (Créer un cluster).

Pour associer, mettre à jour ou supprimer une politique de résiliation automatique sur un cluster en cours d'exécution avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Sélectionnez Clusters et choisissez le cluster que vous souhaitez mettre à jour.
3. Sélectionnez l'onglet Matériel sur la page de détails du cluster.
4. Sélectionnez ou désélectionnez Activer la résiliation automatique pour activer ou désactiver la fonctionnalité. Si vous activez la résiliation automatique, spécifiez le nombre d'heures et de minutes d'inactivité après lequel le cluster doit se résilier automatiquement..

AWS CLI

Avant de commencer

Avant de travailler avec des politiques de résiliation automatique, nous vous recommandons de passer à la dernière version de l' AWS CLI. Pour obtenir des instructions, consultez Installation, mise à jour et désinstallation d' AWS CLI.

Pour joindre ou mettre à jour une politique de résiliation automatique à l'aide de l' AWS CLI

- Vous pouvez utiliser la commande `aws emr put-auto-termination-policy` pour associer ou mettre à jour une politique de résiliation automatique sur un cluster.

  L'exemple suivant indique 3 600 secondes pour *`IdleTimeout`*. Si vous ne le spécifiez pas *`IdleTimeout`*, la valeur par défaut est une heure.

  ```
  aws emr put-auto-termination-policy \
  --cluster-id <your-cluster-id> \
  --auto-termination-policy IdleTimeout=3600
  ```

  > **Note**
  >
  > Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

  Vous pouvez également spécifier une valeur pour `--auto-termination-policy` lorsque vous utilisez la commande `aws emr create-cluster`. Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le AWS CLI, consultez la référence des AWS CLI commandes.

Pour supprimer une politique de résiliation automatique à l'aide du AWS CLI

- Utilisez la commande `aws emr remove-auto-termination-policy` pour supprimer une politique de résiliation automatique d'un cluster. Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le AWS CLI, consultez la référence des AWS CLI commandes.

  ```
  aws emr remove-auto-termination-policy --cluster-id <your-cluster-id>
  ```

Utilisation de la protection contre la résiliation

La protection contre les interruptions protège vos clusters contre les interruptions accidentelles, ce qui peut être particulièrement utile pour les clusters de longue durée traitant des charges de travail critiques. 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. 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 résilier un cluster, une étape supplémentaire vous est proposée pour désactiver la protection contre la résiliation.

> **Warning**
>
> La protection contre la résiliation ne garantit pas la conservation des données en cas d'erreur humaine ou de solution de contournement, par exemple, si une commande de redémarrage est émise depuis la ligne de commande alors que vous êtes connecté à l'instance via SSH, si une application ou un script exécuté sur l'instance émet une commande de redémarrage ou si l'API Amazon EC2 ou Amazon EMR est utilisée pour désactiver la protection contre la résiliation. Cela est également vrai si vous utilisez Amazon EMR versions 7.1 ou supérieures et qu'une instance devient défectueuse et irrécupérable. Même lorsque la protection contre la résiliation est activée, les données enregistrées sur le stockage de l'instance, y compris les données HDFS, peuvent être perdues. Rédigez les données de sortie vers les sites Amazon S3 et créez des stratégies de sauvegarde adaptées à vos exigences de continuité d'activité.

La protection de la résiliation n'affecte pas votre capacité à dimensionner des ressources de cluster en utilisant l'une des actions suivantes :

- Redimensionner un cluster manuellement à l'aide du AWS Management Console ou AWS CLI. Pour plus d’informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution.
- La suppression des instances d'un cœur ou d'un groupe d'instances de tâches à l'aide d'une stratégie de diminution en charge avec le dimensionnement automatique. Pour plus d’informations,

consultez Utilisation de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.

- Suppression des instances d'un parc d'instances en réduisant la capacité cible. Pour plus d'informations, consultez Options de parc d'instances.

Protection contre la résiliation et Amazon EC2

Le paramètre de protection contre la résiliation dans un cluster Amazon EMR correspond à l'`DisableApiTerminationattribut` de toutes les instances Amazon EC2 du cluster. Par exemple, si vous activez la protection contre la résiliation dans un cluster EMR, Amazon EMR définit automatiquement la valeur true `DisableApiTermination` pour toutes les instances EC2 du cluster EMR. Il en va de même si vous désactivez la protection contre le licenciement. Amazon EMR définit automatiquement la valeur false `DisableApiTermination` pour toutes les instances EC2 du cluster EMR. Si vous résiliez ou réduisez un cluster depuis Amazon EMR et que les paramètres Amazon EC2 sont en conflit avec une instance EC2, Amazon EMR donne la priorité au `DisableApiStop` paramètre Amazon EMR par `DisableApiTermination` rapport aux paramètres et dans Amazon EC2 et continue de mettre fin à l'instance EC2.

Par exemple, vous pouvez utiliser la console Amazon EC2 pour activer la protection de résiliation sur une instance Amazon EC2 dans un cluster EMR avec la protection de résiliation désactivée. Si vous résiliez ou réduisez le cluster à l'aide de la console Amazon EMR, de ou de l'API Amazon EMR AWS CLI, Amazon EMR remplace le `DisableApiTermination` paramètre, le définit sur false et met fin à l'instance ainsi qu'aux autres instances.

Vous pouvez également utiliser la console Amazon EC2 pour activer la protection d'arrêt sur une instance Amazon EC2 dans un cluster EMR avec la protection de terminaison désactivée. Si vous résiliez ou réduisez le cluster, Amazon EMR définit la valeur false dans Amazon EC2 et met fin `DisableApiStop` à l'instance ainsi qu'aux autres instances.

Amazon EMR remplace le `DisableApiStop` paramètre uniquement lorsque vous mettez fin à un cluster ou que vous le réduisez. Lorsque vous activez ou désactivez la protection contre la résiliation dans un cluster EMR, Amazon EMR ne modifie le `disableApiStop` paramètre d'aucune des instances EC2 du cluster EMR correspondant.

> **Important**
>
> Si vous créez une instance dans le cadre d'un cluster Amazon EMR avec protection de terminaison, que vous utilisez l'API AWS CLI ou les commandes Amazon EC2 pour modifier

l'instance en conséquence, puis `DisableApiTermination` que `false` l'API AWS CLI ou les commandes Amazon EC2 exécutent l'opération`TerminateInstances`, l'instance Amazon EC2 se termine.

Protection contre la résiliation et nœuds YARN non sains

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 NodeManager de contrôle de santé. Si un nœud est signaléUNHEALTHY, le contrôleur d'instance Amazon EMR l'ajoute à une liste de refus et ne lui alloue pas de conteneurs YARN tant qu'il n'est pas redevenu sain. En fonction de l'état de la protection contre la résiliation, du remplacement du nœud défectueux et de la version publiée d'Amazon EMR, Amazon EMR remplacera l'instance défectueuse ou cessera d'allouer des contrôleurs à l'instance.

Protection contre la résiliation et exécution de la résiliation après une étape

Lorsque vous activez la résiliation après l'exécution des étapes et que vous activez également la protection contre la résiliation, Amazon EMR ignore la protection contre la résiliation.

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 les versions antérieures), `CANCEL_AND_WAIT`, et `CONTINUE`. Pour plus d'informations, consultez Soumission de travail à un cluster.

En cas d'échec d'une étape configurée avec la `ActionOnFailure` valeur définie sur`CANCEL_AND_WAIT`, si l'arrêt après exécution des étapes est activé, le cluster se termine sans exécuter les 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.

ActionOnDéfaillance	Résiliation après exécution de l'étape	Protection de la résiliation	Résultat
	Activé	Désactivées	Le cluster se résilie

ActionOnDéfaillance	Résiliation après exécution de l'étape	Protection de la résiliation	Résultat
`TERMINATE_CLUSTER`	Activées	Activées	Le cluster se résilie
	Désactivées	Activées	Le cluster continue
	Désactivées	Désactivées	Le cluster se résilie

Protection contre la résiliation et instances Spot

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.

Configuration de la protection contre la résiliation lorsque vous lancez un cluster

Vous pouvez activer ou désactiver la protection contre les interruptions lorsque vous lancez un cluster à l'aide de la console AWS CLI, de l'API ou de l'API.

Pour les clusters à nœud unique, les paramètres de protection de terminaison par défaut sont les suivants :

- Lancement d'un cluster par la console Amazon EMR : la protection contre la résiliation est désactivée par défaut.
- Le lancement d'un cluster par AWS CLI `aws emr create-cluster` —Termination Protection est désactivé sauf indication contraire`--termination-protected`.
- Lancement d'un cluster à l'aide de la commande Amazon EMR API RunJobFlow : la protection contre la résiliation est désactivée sauf si la valeur `TerminationProtected` booléenne est définie sur. `true`

Pour les clusters à haute disponibilité, les paramètres de protection de terminaison par défaut sont les suivants :

- Lancement d'un cluster par la console Amazon EMR — La protection contre la résiliation est activée par défaut.
- Le lancement d'un cluster par AWS CLI `aws emr create-cluster` —Termination Protection est désactivé sauf indication contraire`--termination-protected`.

- Lancement d'un cluster à l'aide de la commande Amazon EMR API RunJobFlow : la protection contre la résiliation est désactivée sauf si la valeur `TerminationProtected` booléenne est définie sur. `true`

Console

Pour activer ou désactiver la protection contre la résiliation lorsque vous créez un cluster à l'aide de la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Pour la version d'EMR, choisissez emr-6.6.0 ou une version ultérieure.
4. Sous Résiliation du cluster et remplacement du nœud, assurez-vous que l'option Utiliser la protection de terminaison est présélectionnée, ou désactivez la sélection pour la désactiver.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

AWS CLI

Pour activer ou désactiver la protection contre la résiliation lorsque vous créez un cluster à l'aide du AWS CLI

- Avec le AWS CLI, vous pouvez lancer un cluster avec la protection de terminaison activée à l'aide de la `create-cluster` commande avec le `--termination-protected` paramètre. 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 continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "TerminationProtectedCluster" --release-
label emr-7.1.0 \
--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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Configuration de la protection contre la résiliation pour les clusters en cours d'exécution

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.

Console

Pour activer ou désactiver la protection contre la résiliation pour un cluster en cours d'exécution à l'aide de la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Propriétés de la page des détails du cluster, recherchez Résiliation du cluster et sélectionnez Modifier.
4. Cochez ou décochez la case Utiliser la protection contre la résiliation pour activer ou désactiver la fonctionnalité. Sélectionnez ensuite Enregistrer les modifications pour confirmer.

AWS CLI

Pour activer ou désactiver la protection contre les interruptions pour un cluster en cours d'exécution à l'aide du 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 active la protection de la résiliation sur le cluster doté de l'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
```

Remplacement de nœuds malsains

Amazon EMR utilise régulièrement le service de vérification de l'NodeManager état d'Apache Hadoop pour surveiller l'état des nœuds principaux de votre Amazon EMR sur les clusters Amazon EC2. Si le fonctionnement d'un nœud n'est pas optimal, le vérificateur de santé le signale au contrôleur Amazon EMR. Le contrôleur Amazon EMR ajoute le nœud à une liste de refus, empêchant ainsi le nœud de recevoir de nouvelles applications YARN jusqu'à ce que son état s'améliore. L'une des raisons les plus courantes pour lesquelles un nœud peut devenir insalubre est la surexploitation du disque. Pour plus d'informations sur l'identification des nœuds défectueux et la restauration, consultez la section Erreurs liées aux ressources.

Vous pouvez choisir si Amazon EMR doit mettre fin aux nœuds défectueux ou les conserver dans le cluster. Si vous désactivez le remplacement de nœuds défectueux, les nœuds défectueux restent dans la liste de refus et continuent à être pris en compte dans la capacité du cluster. Vous pouvez toujours vous connecter à votre instance principale Amazon EC2 pour la configuration et la restauration, afin de pouvoir redimensionner votre cluster pour augmenter la capacité. Notez qu'Amazon EMR remplacera les nœuds défectueux même si la protection contre la résiliation est activée.

Si le remplacement de nœuds défectueux est activé, Amazon EMR mettra fin au nœud principal défectueux et fournira une nouvelle instance en fonction du nombre d'instances du groupe d'instances ou de la capacité cible pour les flottes d'instances. Si plusieurs nœuds principaux ou tous

les nœuds principaux ne fonctionnent pas correctement pendant plus de 45 minutes, Amazon EMR remplacera les nœuds avec élégance.

> **Important**
>
> Pour éviter le risque de perdre définitivement des données HDFS car Amazon EMR remplace gracieusement une instance principale défectueuse, nous vous recommandons de toujours sauvegarder vos données.

Amazon EMR publie Amazon CloudWatch Events pour le remplacement de nœuds défectueux, afin que vous puissiez suivre l'évolution de vos instances principales défaillantes. Pour plus d'informations, consultez la section Événements de remplacement de nœuds défectueux.

Paramètres de protection par défaut pour le remplacement et la terminaison des nœuds

Le remplacement de nœuds défectueux est disponible pour toutes les versions d'Amazon EMR, mais les paramètres par défaut dépendent du label de version que vous choisissez. Vous pouvez modifier n'importe lequel de ces paramètres en configurant le remplacement de nœuds défectueux lors de la création d'un nouveau cluster ou en accédant à la configuration du cluster à tout moment.

Si vous créez un cluster à nœud unique ou un cluster à haute disponibilité exécutant Amazon EMR version 7.0 ou antérieure, le paramètre par défaut de remplacement de nœud défectueux dépend de la protection contre la résiliation :

- L'activation de la protection de terminaison désactive le remplacement de nœuds défectueux.
- La désactivation de la protection de terminaison entraîne le remplacement d'un nœud défectueux.

Configuration du remplacement de nœuds défectueux lorsque vous lancez un cluster

Vous pouvez activer ou désactiver le remplacement de nœuds défectueux lorsque vous lancez un cluster à l'aide de la console, de l' AWS CLI API ou de l'API.

Le paramètre de remplacement des nœuds défectueux par défaut dépend de la manière dont vous lancez le cluster :

- Console Amazon EMR : le remplacement de nœuds défectueux est activé par défaut.
- AWS CLI `aws emr create-cluster`— le remplacement de nœuds défectueux est activé par défaut, sauf indication contraire de votre part`--no-unhealthy-node-replacement`.
- Commande d'RunJobFlow API Amazon EMR : le remplacement de nœuds défectueux est activé par défaut, sauf si vous définissez la valeur `UnhealthyNodeReplacement` booléenne sur ou. `True` `False`

Console

Pour activer ou désactiver le remplacement de nœuds défectueux lorsque vous créez un cluster avec la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Pour la version EMR, choisissez le label de version Amazon EMR que vous souhaitez.
4. Sous Résiliation du cluster et remplacement du nœud, assurez-vous que le remplacement du nœud défectueux (recommandé) est présélectionné, ou désactivez la sélection pour le désactiver.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

AWS CLI

Pour activer ou désactiver le remplacement de nœuds défectueux lorsque vous créez un cluster à l'aide du AWS CLI

- Avec le AWS CLI, vous pouvez lancer un cluster avec le remplacement de nœuds défectueux activé à l'aide de la `create-cluster` commande avec le `--unhealthy-node-replacement` paramètre. Le remplacement des nœuds défectueux est activé par défaut.

  L'exemple suivant crée un cluster sur lequel le remplacement de nœuds défectueux est activé :

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "SampleCluster" --release-label emr-7.1.0 \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --unhealthy-node-replacement
```

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le AWS CLI, consultez la section Commandes Amazon EMR. AWS CLI

Configuration du remplacement de nœuds défectueux dans un cluster en cours d'exécution

Vous pouvez activer ou désactiver le remplacement de nœuds défectueux pour un cluster en cours d'exécution à l'aide de la console AWS CLI, de l'API ou de l'API.

Console

Pour activer ou désactiver le remplacement de nœuds défectueux pour un cluster en cours d'exécution avec la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Propriétés de la page des détails du cluster, recherchez Résiliation du cluster et remplacement de nœuds, puis sélectionnez Modifier.
4. Cochez ou décochez la case Remplacement du nœud défectueux pour activer ou désactiver la fonctionnalité. Sélectionnez ensuite Enregistrer les modifications pour confirmer.

AWS CLI

Pour activer ou désactiver le remplacement de nœuds défectueux pour un cluster en cours d'exécution à l'aide du AWS CLI

- Pour activer le remplacement de nœuds défectueux sur un cluster en cours d'exécution avec le AWS CLI, utilisez la `modify-cluster-attributes` commande avec le `--unhealthy-node-replacement` paramètre. Pour la désactiver, utilisez le paramètre `--no-unhealthy-node-replacement`.

  L'exemple suivant active le remplacement de nœuds défectueux sur le cluster portant l'ID *J-3KVTxxxxxx7UG* :

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --unhealthy-node-replacement
  ```

  L'exemple suivant désactive le remplacement de nœuds défectueux sur le même cluster :

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-unhealthy-node-replacement
  ```

Utilisation des AMI Amazon Linux dans Amazon EMR

Amazon Linux Amazon Machine Images (AMI)

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. Pour plus d'informations sur l'AMI Amazon Linux par défaut, consultez Utilisation de l'AMI Amazon Linux par défaut pour Amazon EMR. 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. Vous pouvez spécifier une AMI personnalisée pour chaque type d'instance

dans la configuration du groupe d'instances ou du parc d'instances d'un cluster Amazon EMR. La prise en charge de plusieurs AMI personnalisées vous donne la possibilité d'utiliser plusieurs types d'architecture dans un cluster. veuillez consulter Utilisation d'une image AMI personnalisée.

Amazon EMR attache automatiquement un volume SSD d'usage général Amazon EBS en tant que dispositif racine pour toutes les AMI. Les AMI basées sur EBS améliorent les performances. Pour en savoir plus sur les AMI Amazon Linux, consultez Amazon Machine Images (AMI). Pour plus d'informations sur le stockage d'instance pour les instances Amazon EMR, consultez Stockage d'instances.

Utilisation de l'AMI Amazon Linux par défaut pour Amazon EMR

Chaque version d'Amazon EMR utilise une AMI Amazon Linux par défaut pour Amazon EMR, sauf si vous spécifiez une AMI personnalisée. À partir des versions Amazon EMR 5.36, Amazon EMR 6.6 et Amazon EMR 7.0, le comportement par défaut pour mettre à jour Amazon Linux 2 (AL2 pour EMR 5.x et 6.x, AL2023 pour EMR 7.x) dans une AMI par défaut Amazon EMR consiste à appliquer automatiquement la dernière version d'Amazon Linux à l'AMI Amazon EMR par défaut.

Mises à jour automatiques d'Amazon Linux pour les versions d'Amazon EMR

Lorsque vous lancez un cluster avec le dernier correctif d'Amazon EMR version 7.0 ou ultérieure, version 6.6 ou ultérieure ou version 5.36 ou ultérieure, Amazon EMR utilise la dernière version d'Amazon Linux pour l'AMI Amazon EMR par défaut. Par exemple :

- Lorsqu'il existe une version `x.x.0` et `x.x.1`, `x.x.0` cesse de recevoir les mises à jour de l'AMI lors du lancement de `x.x.1`.
- De même, `x.x.1` cesse de recevoir les mises à jour de l'AMI lors des lancements de `x.x.2`.
- Plus tard, lors de la sortie de `x.y.0`, `x.x.[latest]` continue de recevoir les mises à jour de l'AMI en même temps que `x.y.[latest]`.

Pour savoir si vous utilisez la dernière version du correctif, comme indiqué par le chiffre après la deuxième décimale (`6.8.1`) pour une version d'Amazon EMR, reportez-vous aux versions disponibles dans le Guide de version Amazon EMR, consultez le menu déroulant des Versions Amazon EMR lorsque vous créez un cluster dans la console, ou utilisez l'API `ListReleaseLabels` ou l'action de la CLI `list-release-labels`. Pour recevoir des mises à jour lorsque nous lançons une nouvelle version d'Amazon EMR, abonnez-vous au flux RSS sur la page Quoi de neuf ? du Guide de publication.

Si vous le souhaitez, vous pouvez choisir de lancer votre cluster avec la version Amazon Linux fournie pour la première fois avec la version Amazon EMR. Pour plus d'informations sur la manière de spécifier la version Amazon Linux pour votre cluster, consultez Modification de la version d'Amazon Linux lorsque vous créez un cluster EMR.

Versions Amazon Linux par défaut

Rubriques

- AMI par défaut pour Amazon EMR 7.0 et versions ultérieures
- AMI par défaut pour Amazon EMR 6.6 et versions ultérieures
- AMI par défaut pour Amazon EMR 5.x

AMI par défaut pour Amazon EMR 7.0 et versions ultérieures

Le tableau suivant répertorie les informations Amazon Linux relatives à la dernière version du correctif des versions 7.0 et supérieures d'Amazon EMR.

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023,3.2 240304,(	6.1.79-99,164 mai 2023	12 mars 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023,3.2 240219,(	6.1.77-99,164 mai 2023	1er mars 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023,3.2 240205,(	6,175-99,163 mai 2023	19 février 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023,3.2 240122.(	6,172-96,166 mai 2023	5 février 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023,3.2 240108.(	6,172-96,166 mai 2023	24 janvier 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2023.3.2 23 1211,4	6.1.66-91.160.amzn2023	19 décembre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-south-1 • ca-central-1 • il-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

AMI par défaut pour Amazon EMR 6.6 et versions ultérieures

Le tableau suivant répertorie les informations Amazon Linux relatives à la dernière version du correctif d'Amazon EMR 6.6.x et versions ultérieures.

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2024 223.0	4,14.336	8 mars 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `eu-central-2` (6,1,1 et versions ultérieures) • `eu-south-1` • `eu-south-2` (6,1,1 et versions ultérieures) • `ap-east-1` • `ap-south-1` • `ap-south-2` (6,1,1 et versions ultérieures) • `ap-southeast-3` • `ap-southe ast-4` (6.8.1+ et 5.36.1) • `ap-northeast-1` • `ap-northeast-2` • `ap-northeast-3` • `ap-southeast-1` • `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (6,1,1 et versions ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (6.8.1+ et 5.36.1)

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ca-west-1` (6.9.1+ et 5.36.1) • `us-gov-east-1` • `us-gov-west-1` • `cn-north-1` • `cn-northeast-1`

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2024 131.0	4,14.336	14 février 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1,1 et versions ultérieures) • eu-south-1 • eu-south-2 (6,1,1 et versions ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (6,1,1 et versions ultérieures) • ap-southeast-3 • ap-southeast-4 (6.8.1+ et 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6,1,1 et versions ultérieures) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ et 5.36.1) • ca-west-1 (6.9.1+ et 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2024 124.0	4,14.336	7 février 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1,1 et versions ultérieures) • eu-south-1 • eu-south-2 (6,1,1 et versions ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (6,1,1 et versions ultérieures) • ap-southeast-3 • ap-southeast-4 (6.8.1+ et 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (6,1,1 et versions ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (6.8.1+ et 5.36.1) • `ca-west-1` (6.9.1+ et 5.36.1) • `us-gov-east-1` • `us-gov-west-1` • `cn-north-1` • `cn-northeast-1`

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2024 109.0	4,1,4334	24 janvier 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1,1 et versions ultérieures) • eu-south-1 • eu-south-2 (6,1,1 et versions ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (6,1,1 et versions ultérieures) • ap-southeast-3 • ap-southeast-4 (6.8.1+ et 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6,1,1 et versions ultérieures) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ et 5.36.1) • ca-west-1 (6.9.1+ et 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 218,0	4,14,330	2 janvier 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 206,0	4,14,330	22 décembre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 116,0	4,14,328	11 décembre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 101,0	4,1,4327	17 novembre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 020.1	4,1,4326	7 novembre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 012.1	4,1,4326	26 octobre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 926,0	4,14,322	19 octobre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.8 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 8906.0	4,14,322	4 octobre 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieur es, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-northeast-3` • `ap-southeast-1` • `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (version 6.9 et ultérieures, et version 5.36.1)

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 822.0	4,14,322	30 août 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-northeast-3` • `ap-southeast-1` • `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (version 6.9 et ultérieures, et version 5.36.1)

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 808.0	4,14,320	24 août 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (version 6.10 et ultérieures) • me-south-1 • ca-central-1 • il-central-1 (version 6.9 et ultérieures, et version 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 727,0	4,14,320	14 août 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-northeast-3` • `ap-southeast-1` • `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (version 6.9 et ultérieures, et version 5.36.1)

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 719,0	4,14,320	02/08/2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-southeast-4 (version 6.8 et ultérieures, et version 5.36.1) • ap-northeast-1 • ap-northeast-2

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-northeast-3` • `ap-southeast-1` • `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1` • `il-central-1` (version 6.9 et ultérieures, et version 5.36.1)

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 628,0	4,14,318	12 juillet 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1`

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 612,0	4,1,4314	23 juin 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieures) • eu-south-1 • eu-south-2 (version 6.10 et ultérieures) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieures) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ap-southeast-2` • `af-south-1` • `sa-east-1` • `me-central-1` (version 6.10 et ultérieures) • `me-south-1` • `ca-central-1`

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2023 504.1	4,14.313	16 mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 et ultérieur es) • eu-south-1 • eu-south-2 (version 6.10 et ultérieur es) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 et ultérieur es) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 418,0	4,14.311	3 mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (version 6.10 uniquement) • eu-south-1 • eu-south-2 (version 6.10 uniquement) • ap-east-1 • ap-south-1 • ap-south-2 (version 6.10 uniquement) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2023 404.1	4,14.311	18 avril 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ca-central-1
2,0.2023 404.0	4,14.311	10 avril 2023	• us-east-1 • eu-west-3

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 320,0	4,14,309	30 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• `ca-central-1`

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 307.0	4,14,305	15 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 207.0	4,14,304	3 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• ca-central-1
2.0.2023 119.1	4,14,301	9 février 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2022 210.1	4,14,301	12 janvier 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 103.3	4,14.296	5 décembre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 004.0	4,14.294	2 novembre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 912.1	4,14,291	7 octobre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1
2,0,2022 805.0	4,14.287	30 août 2022	• us-west-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2022 719,0	4,14.287	10 août 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 426.0	4,14,281	10 juin 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 406.1	4,14,275	2 mai 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

AMI par défaut pour Amazon EMR 5.x

Le tableau suivant répertorie les informations Amazon Linux relatives à la dernière version du correctif d'Amazon EMR 5.x, versions 5.36 et supérieures.

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2023 504.1	4,14.313	16 mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
			• me-central-1
2,0,2023 418,0	4,14.311	3 mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • me-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2023 404.1	4,14.311	18 avril 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1
2,0.2023 404.0	4,14.311	10 avril 2023	• us-east-1 • eu-west-3

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2023 320,0	4,14,309	30 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 307.0	4,14,305	15 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2023 207.0	4,14,304	3 mars 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2.0.2022 210.1	4,14,301	12 janvier 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 103.3	4,14.296	5 décembre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 004.0	4,14.294	2 novembre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 912.1	4,14,291	7 octobre 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0,2022 719,0	4,14.287	10 août 2022	• us-west-1 • eu-west-3 • eu-north-1 • eu-central-1 • ap-south-1 • me-south-1

OsRelea Label (Toutes les versions)	Version de noyau AL	Date de disponibilité	Régions AWS
2,0.2022 426.0	4,14,281	14 juin 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

Considérations relatives aux mises à jour

Notez les comportements par défaut des mises à jour logicielles :

Amazon EMR 7.x – Amazon Linux 2023

La version 7.0 et les versions ultérieures d'Amazon EMR s'exécutent sur Amazon Linux 2023 (AL2023). Le comportement par défaut pour AL2023 consiste à verrouiller les AMI sur une version spécifique du référentiel logiciel Amazon Linux. Par conséquent, les mises à jour de sécurité ne sont pas appliquées chaque fois que vous lancez un cluster. En revanche, le comportement par défaut pour les versions 7.x d'Amazon EMR consiste à appliquer automatiquement la dernière version d'AL2023 à l'AMI Amazon EMR uniquement lorsque vous créez le cluster. Pour recevoir les dernières mises à jour de sécurité, nous vous recommandons de recréer régulièrement votre cluster.

Amazon EMR 5.x et 6.x – Amazon Linux et Amazon Linux 2

Pour les versions d'Amazon EMR antérieures à la version 7.0, lorsqu'une instance Amazon EC2 se lance pour la première fois dans un cluster basé sur l'AMI Amazon Linux (AL) ou Amazon Linux 2 (AL2) par défaut, elle recherche les mises à jour logicielles applicables à la version de lancement dans les référentiels de packages activés pour AL et Amazon EMR. Comme avec les autres instances AL et AL2, les mises à jour de sécurité critiques et importantes provenant de ces référentiels sont automatiquement installées.

Notez également que votre configuration réseau doit autoriser la sortie HTTP et HTTPS vers les référentiels Amazon Linux dans Amazon S3, sinon, les mises à jour de sécurité échoueront. Pour plus d'informations, consultez Amazon Linux - Package repository dans le guide de l'utilisateur Amazon EC2. Par défaut, les autres packages logiciels et mises à jour du noyau nécessitant un redémarrage, notamment NVIDIA et CUDA, sont exclus du téléchargement automatique au premier démarrage.

Amazon EMR 5.35.0 et versions antérieures, et 6.5.0 et versions antérieures : l'AMI Amazon Linux est « verrouillée » sur la version de lancement d'Amazon EMR

Pour Amazon EMR 5.35.0 et versions antérieures, et 6.5.0 et versions antérieures, l'AMI par défaut est basée sur la plupart des AMI Amazon up-to-date Linux disponibles 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 d'Amazon EMR 5.35.0 et versions antérieures, et 6.5.0 et versions antérieures d'Amazon EMR est « verrouillée » sur la version d'AMI Amazon Linux qui lui est attribuée afin de garantir la compatibilité. Pour cette raison, nous vous conseillons d'utiliser la version d'Amazon EMR la plus récente, sauf si vous avez besoin d'une version inférieure pour la compatibilité et que vous n'êtes pas en mesure de migrer. Si vous devez utiliser une version inférieure d'Amazon EMR pour

des raisons de compatibilité, nous vous recommandons d'utiliser la dernière version 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.

Pour plus d'informations sur le comportement de mise à jour automatique introduit avec Amazon EMR 5.36.0 et versions ultérieures et 6.6.0 et versions ultérieures, consultez Mises à jour automatiques d'Amazon Linux pour les versions d'Amazon EMR.

Le comportement de démarrage par défaut exclut les mises à jour du noyau

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 de sécurité critiques et importantes provenant de ces référentiels sont automatiquement installées.

Toutefois, si vous utilisez une ancienne version de l'AMI Amazon Linux, il est possible que la dernière mise à jour de sécurité ne soit pas installée automatiquement. Cela est dû au fait que les référentiels référencés par votre cluster EMR sont fixes pour chaque version de l'AMI Amazon Linux.

Notez également que votre configuration réseau doit autoriser la sortie HTTP et HTTPS vers les référentiels Amazon Linux dans Amazon S3, sinon, les mises à jour de sécurité échoueront. Pour plus d'informations, consultez Amazon Linux - Package repository dans le guide de l'utilisateur Amazon EC2. Par défaut, les autres packages logiciels et mises à jour du noyau nécessitant un redémarrage, notamment NVIDIA et CUDA, sont exclus du téléchargement automatique au premier démarrage.

> **Important**
>
> Les clusters EMR qui exécutent AL2023 utilisent le comportement par défaut d'Amazon Linux, et vos Amazon Machine Images (AMI) sont verrouillées sur une version spécifique du référentiel Amazon Linux. Par défaut, vos clusters ne recevront pas automatiquement les mises à jour de sécurité logicielles au lancement. Vos clusters contiennent uniquement les mises à jour disponibles dans la version de l'AMI AL2023 que vous avez choisie lors de la création de votre cluster. Pour plus d'informations, voir la rubrique Mise à jour d'Amazon Linux 2023 du Guide de l'utilisateur Amazon Linux 2023.

> **Important**
>
> Les clusters Amazon EMR qui exécutent des AMI (Amazon Machine Images) Amazon Linux ou Amazon Linux 2 utilisent le comportement par défaut d'Amazon Linux et ne téléchargent pas et n'installent pas automatiquement les mises à jour importantes et critiques du noyau nécessitant un redémarrage. Ce comportement est identique à celui des autres instances Amazon EC2 qui exécutent l'AMI Amazon Linux par défaut. Si de nouvelles mises à jour logicielles Amazon Linux nécessitant un redémarrage (telles que les mises à jour du noyau, de NVIDIA et de CUDA) sont disponibles après la publication d'une version d'Amazon EMR, les instances de cluster EMR qui exécutent l'AMI par défaut ne téléchargent pas et n'installent pas automatiquement ces mises à jour. Pour obtenir les mises à jour du noyau, vous pouvez personnaliser votre AMI Amazon EMR afin d'utiliser la dernière AMI Amazon Linux.

Le cluster est lancé avec ou sans mises à jour

Sachez que si les mises à jour logicielles ne peuvent pas être installées parce que les référentiels de packages sont inaccessibles au premier démarrage du cluster, l'instance de cluster termine tout de même son lancement. Par exemple, les référentiels peuvent être inaccessibles car S3 est temporairement indisponible, ou vous avez peut-être configuré des règles de VPC ou de pare-feu pour bloquer l'accès.

N'exécutez pas **`sudo yum update`**

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 recommandons vivement de ne pas exécuter `sudo yum update` sur les instances de cluster, que ce soit lorsque vous êtes connecté avec SSH ou lorsque vous utilisez une action de démarrage. Cela pourrait entraîner des erreurs de compatibilité, car tous les packages sont installés sans distinction.

Bonnes pratiques en matière de mise à jour logicielle

Bonnes pratiques pour la gestion des mises à jour logicielles

- Si vous utilisez une version inférieure d'Amazon EMR, envisagez et testez une migration vers 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 clonage des clusters avec la console Amazon EMR est utile à cet effet.
- 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 les 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 Création d'actions d'amorçage pour installer des logiciels supplémentaires.

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.
- 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 d'Amazon EMR inférieure à 5.24.0. Comme pour l’AMI par défaut, la taille minimale du volume racine pour une AMI personnalisée est de 10 Gio pour Amazon EMR 6.9 et versions ultérieures, et de 15 Gio pour Amazon EMR 6.10 et versions ultérieures. Pour

plus d'informations, consultez Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré.

> **Note**
>
> À partir de la version 5.24.0 d'Amazon EMR, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous le spécifiez comme fournisseur de clés. AWS KMS Pour plus d'informations, consultez Chiffrement de disque local.

Une AMI personnalisée doit exister dans la même AWS région que celle dans laquelle vous créez le cluster. Il doit également correspondre à l'architecture de l'instance EC2. Par exemple, une instance m5.xlarge possède une architecture x86_64. Par conséquent, pour provisionner un m5.xlarge à l'aide d'une AMI personnalisée, votre AMI personnalisée doit également avoir une architecture x86_64. De même, pour provisionner une instance m6g.xlarge dotée d'une architecture arm64, votre AMI personnalisée doit avoir une architecture arm64. Pour plus d'informations sur l'identification d'une AMI Linux pour votre type d'instance, consultez Trouver une AMI Linux dans le guide de l'utilisateur Amazon EC2.

> **Important**
>
> Les clusters EMR qui exécutent des AMI (Amazon Linux Machine Images) Amazon Linux ou Amazon Linux 2 utilisent le comportement par défaut d'Amazon Linux et ne téléchargent pas et n'installent pas automatiquement les mises à jour importantes et critiques du noyau nécessitant un redémarrage. Ce comportement est identique à celui des autres instances Amazon EC2 qui exécutent l'AMI Amazon Linux par défaut. Si de nouvelles mises à jour logicielles Amazon Linux nécessitant un redémarrage (telles que les mises à jour du noyau, de NVIDIA et de CUDA) sont disponibles après la publication d'une version d'Amazon EMR, les instances de cluster EMR qui exécutent l'AMI par défaut ne téléchargent pas et n'installent pas automatiquement ces mises à jour. Pour obtenir les mises à jour du noyau, vous pouvez personnaliser votre AMI Amazon EMR afin d'utiliser la dernière AMI Amazon Linux.

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.

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 le AWS CLI pour exécuter la commande suivante, qui crée une instance à partir d'une AMI existante. *MyKeyName*Remplacez-le par la paire de clés que vous utilisez pour vous connecter à l'instance et l'*MyAmiidentifiant* 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 continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

   ```
   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 section Connexion à votre instance Linux à l'aide de SSH dans le guide de l'utilisateur Amazon EC2.
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 plus d'informations, consultez Utiliser une seule AMI personnalisée dans un cluster EMR et Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré.

Comment utiliser une AMI personnalisée dans un cluster Amazon EMR

Vous pouvez utiliser une AMI personnalisée pour provisionner un cluster Amazon EMR de deux manières :

- Utilisez une seule AMI personnalisée pour toutes les instances EC2 du cluster.
- Utilisez différentes AMI personnalisées pour les différents types d'instances EC2 utilisés dans le cluster.

Vous ne pouvez utiliser qu'une des deux options lors du provisionnement d'un cluster EMR, et vous ne pouvez pas le modifier une fois que le cluster a démarré.

Considérations relatives à l'utilisation d'une ou de plusieurs AMI personnalisées dans un cluster Amazon EMR

Considération	AMI personnalisée unique	Plusieurs AMI personnalisées
Utiliser à la fois des processeurs x86 et Graviton2 avec des AMI personnalisées dans le même cluster	✗ Non pris en charge	✓ Pris en charge
La personnalisation de l'AMI varie selon le type d'instance	✗ Non pris en charge	✓ Pris en charge
Modifiez les AMI personnalisées lors de l'ajout de nouveaux groupes/parcs d'instances de tâches à un cluster en cours d'exécution. Remarque : vous ne pouvez pas modifier l'AMI personnalisée des groupes/parcs d'instances existants.	✗ Non pris en charge	✓ Pris en charge
Utiliser AWS la console pour démarrer un cluster	✓ Pris en charge	✗ Non pris en charge
AWS CloudFormation À utiliser pour démarrer un cluster	✓ Pris en charge	✓ Pris en charge

Utiliser une seule AMI personnalisée dans un cluster EMR

Pour spécifier un ID d'AMI personnalisé lorsque vous créez un cluster, utilisez l'une des options suivantes :

- AWS Management Console
- AWS CLI

- Kit SDK Amazon EMR
- Flux d'API Amazon EMR RunJob
- AWS CloudFormation (voir la `CustomAmiID` propriété dans Cluster InstanceGroupConfig, Cluster InstanceTypeConfig InstanceGroupConfig, Resource ou Resource InstanceFleetConfig - InstanceType Config)

Amazon EMR console

Pour spécifier une AMI personnalisée unique dans la console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Nom et applications, recherchez Options du système d'exploitation. Choisissez AMI personnalisée, puis entrez votre ID d'AMI dans le champ AMI personnalisé.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer cluster, choisissez Créer un cluster.

AWS CLI

Pour spécifier une seule AMI personnalisée à l'aide du 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 unique avec un volume de démarrage de 20 Gio. Pour plus d'informations, consultez Personnalisation du volume du périphérique racine Amazon EBS.

  > **Note**
  >
  > Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
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
```

Utiliser plusieurs AMI personnalisées dans un cluster Amazon EMR

Pour créer un cluster à l'aide de plusieurs AMI personnalisées, utilisez l'une des options suivantes :

- AWS CLI version 1.20.21 ou supérieure
- AWS SDK
- Amazon EMR RunJobFlow dans la référence d'API Amazon EMR
- AWS CloudFormation (voir la `CustomAmiID` propriété dans Cluster InstanceGroupConfig, Cluster InstanceTypeConfig InstanceGroupConfig, Resource ou Resource InstanceFleetConfig - InstanceType Config)

La console AWS de gestion ne prend actuellement pas en charge la création d'un cluster à l'aide de plusieurs AMI personnalisées.

Example - Utilisez la AWS CLI pour créer un cluster de groupes d'instances à l'aide de plusieurs AMI personnalisées

À l'aide de la version 1.20.21 ou supérieure de la AWS CLI, vous pouvez attribuer une seule AMI personnalisée à l'ensemble du cluster, ou vous pouvez attribuer plusieurs AMI personnalisées à chaque nœud d'instance de votre cluster.

L'exemple suivant montre un cluster de groupes d'instances uniforme créé avec deux types d'instances (m5.xlarge) utilisés sur tous les types de nœuds (nœuds primaires, principaux et de tâches). Chaque nœud possède plusieurs AMI personnalisées. L'exemple illustre plusieurs fonctionnalités de la configuration personnalisée multiple d'AMI :

- Aucune AMI personnalisée n'est attribuée au niveau du cluster. Cela permet d'éviter les conflits entre les multiples AMI personnalisées et une seule AMI personnalisée, ce qui entraînerait l'échec du lancement du cluster.

- Le cluster peut avoir plusieurs AMI personnalisées sur les nœuds primaires, les nœuds principaux et les nœuds de tâches individuelles. Cela permet de personnaliser les AMI individuelles, telles que les applications préinstallées, les configurations de cluster sophistiquées et les volumes chiffrés du périphérique racine Amazon EBS.
- Le nœud principal du groupe d'instances ne peut avoir qu'un seul type d'instance et l'AMI personnalisée correspondante. De même, le nœud primaire ne peut avoir qu'un seul type d'instance et l'AMI personnalisée correspondante.
- Le cluster peut comporter plusieurs nœuds de tâches.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567
InstanceGroupType=TASK,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-456789
```

Example - Utilisez la version 1.20.21 ou supérieure de la AWS CLI pour ajouter un nœud de tâches à un cluster de groupes d'instances en cours d'exécution avec plusieurs types d'instances et plusieurs AMI personnalisées

À l'aide de la version 1.20.21 ou supérieure de la AWS CLI, vous pouvez ajouter plusieurs AMI personnalisées à un groupe d'instances que vous ajoutez à un cluster en cours d'exécution. L'argument `CustomAmiId` peut être utilisé avec la commande `add-instance-groups` comme le montre l'exemple suivant. Notez que le même ID d'AMI personnalisé multiple (ami-123456) est utilisé dans plusieurs nœuds.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567

{
    "ClusterId": "j-123456",
    ...
}

aws emr add-instance-groups --cluster-id j-123456 --instance-groups
 InstanceGroupType=Task,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
```

Example - Utilisez la version 1.20.21 ou supérieure de la AWS CLI pour créer un cluster de flotte d'instances, plusieurs AMI personnalisées, plusieurs types d'instances, une instance principale à la demande, un cœur à la demande, un cœur à la demande, plusieurs cœurs et nœuds de tâches

```
aws emr create-cluster --instance-fleets
InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-567890}']
```

Example - Utilisez la version 1.20.21 ou supérieure de la AWS CLI pour ajouter des nœuds de tâches à un cluster en cours d'exécution avec plusieurs types d'instances et plusieurs AMI personnalisées

```
aws emr create-cluster --instance-fleets
InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']

{
    "ClusterId": "j-123456",
    ...
}

aws emr add-instance-fleet --cluster-id j-123456 --instance-fleet
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
```

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. Par défaut, les mises à jour du noyau Amazon Linux et les autres packages logiciels nécessitant un redémarrage ne sont pas mis à jour. Notez que votre configuration réseau doit permettre la sortie HTTP et HTTPS vers les référentiels Amazon Linux dans Amazon S3, sinon les mises à jour de sécurité n'aboutiront pas.

> **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.

Avec le AWS Management Console, vous pouvez sélectionner l'option permettant de désactiver les mises à jour lorsque vous choisissez une AMI personnalisée.

Avec le AWS CLI, vous pouvez spécifier `--repo-upgrade-on-boot NONE --custom-ami-id` lorsque vous utilisez la create-cluster commande.

Avec l'API Amazon EMR, vous pouvez spécifier le `NONE` RepoUpgradeOnBootparamètre.

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 chiffrés, consultez la section relative au chiffrement Amazon EBS dans le guide de l'utilisateur Amazon EC2. L'AMI source de l'instantané peut être l'AMI Amazon Linux de base, ou vous pouvez copier un instantané à partir d'une AMI dérivée de l'AMI Amazon Linux de base que vous avez personnalisée.

> **Note**
>
> À partir de la version 5.24.0 d'Amazon EMR, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous le spécifiez comme fournisseur de clés. AWS KMS Pour plus d'informations, consultez Chiffrement de disque local.

Vous pouvez utiliser un fournisseur de clé externe ou une clé AWS KMS pour chiffrer le volume racine EBS. 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 avec l'AMI. Lors de l'utilisation en AWS KMS tant que fournisseur de clés, 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 une clé KMS

La première étape de cet exemple est de trouver l'ARN d'un KMS ou d'en créer une nouvelle. Pour plus d'informations sur la création de clés, consultez Création de clés dans le Guide du développeur AWS Key Management Service . 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 supérieur, lorsque vous créez l'AMI.

Ajout du rôle du service pour Amazon EC2 à la liste des utilisateurs de clés de chiffrement avec la console

1. Connectez-vous à la console AWS Key Management Service (AWS KMS) AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Choisissez l'alias de la clé KMS 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.

Pour créer une AMI chiffrée à l'aide du AWS CLI

- Utilisez la `aws ec2 copy-image` commande du 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 créée ou modifiée plus bas.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
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 Utiliser une seule AMI personnalisée dans un cluster EMR. 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.

Bonnes pratiques et considérations

Lorsque vous créez une AMI personnalisée pour Amazon EMR, soyez conscient de ce qui suit :

- La série Amazon EMR 7.x est basée sur Amazon Linux 2023. Pour ces versions d'Amazon EMR, vous devez utiliser des images basées sur Amazon Linux 2023 pour les AMI personnalisées. Pour trouver une AMI personnalisée de base, consultez Recherche d'une AMI Linux.
- Pour les versions d'Amazon EMR inférieures à 7.x, les AMI Amazon Linux 2023 ne sont pas prises en charge.
- Amazon EMR 5.30.0 et versions ultérieures, ainsi que la série Amazon EMR 6.x sont basés sur Amazon Linux 2. Pour ces versions Amazon 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 inférieures à 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. Une AMI 32 bits n'est pas prise en charge.
- 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'instance pris en charge.
- 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 Amazon Linux personnalisée basée sur une AMI Amazon Linux dont la date de création est le 11/08/2018, le serveur Oozie ne démarre pas. Si vous utilisez Oozie, créez une AMI personnalisée basée sur un ID d'AMI Amazon Linux avec une date de création différente. Vous pouvez utiliser la AWS CLI commande suivante pour renvoyer une liste d'identifiants d'image pour toutes les AMI HVM Amazon Linux avec une version 2018.03, ainsi que la date de sortie, afin de pouvoir choisir une AMI Amazon Linux appropriée comme base. MyRegion Remplacez-le par votre identifiant de région, tel que us-west-2.

  ```
  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
  ```

- Dans les cas où vous utilisez un VPC avec un nom de domaine et un AmazonProvided DNS non standard, vous ne devez pas utiliser `rotate` cette option dans la configuration DNS des systèmes d'exploitation.

Pour plus d'informations, consultez la section Création d'une AMI Linux basée sur Amazon EBS dans le guide de l'utilisateur Amazon EC2.

Modification de la version d’Amazon Linux lorsque vous créez un cluster EMR

Lorsque vous lancez un cluster à l'aide d'Amazon EMR 6.6.0 ou supérieur, il utilise automatiquement la dernière version d'Amazon Linux 2 qui a été validée pour l'AMI Amazon EMR par défaut. Vous pouvez spécifier une version Amazon Linux différente pour votre cluster à l'aide de la console Amazon EMR ou de l' AWS CLI.

Amazon EMR console

Pour modifier la version d’Amazon Linux lorsque vous créez un cluster dans la console

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

2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Pour Version EMR, choisissez emr-6.6.0 ou supérieur.
4. Sous Options du système d'exploitation, choisissez Version Amazon Linux, puis cochez la case Appliquer automatiquement les dernières mises à jour d'Amazon Linux.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

AWS CLI

Pour modifier la version d'Amazon Linux lorsque vous créez un cluster avec AWS CLI

- Utilisez le paramètre `--os-release-label` pour spécifier la Version Amazon Linux lorsque vous exécutez la commande `aws emr` create-cluster.

```
aws emr create-cluster --name "Cluster with Different Amazon Linux Release" \
--os-release-label 2.0.20210312.1 \
--release-label emr-6.6.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

Personnalisation du volume du périphérique racine Amazon EBS

Configuration par défaut du volume racine EBS

Avec Amazon EMR 4.x et versions ultérieures, vous pouvez spécifier la taille du volume racine lorsque vous créez un cluster. Avec Amazon EMR 6.15.0 et versions ultérieures, vous pouvez également spécifier les IOPS et le débit du volume racine. Les attributs s'appliquent uniquement au volume du périphérique racine Amazon EBS et à toutes les instances du cluster. Ils ne s'appliquent pas aux volumes de stockage, que vous spécifiez séparément pour chaque type d'instance lorsque vous créez votre cluster.

- La taille du volume racine par défaut de 15 Gio dans Amazon EMR 6.10.0 et versions ultérieures. Les versions antérieures ont une taille de volume racine par défaut de 10 Gio. Vous pouvez définir une taille maximale de 100 GiB.
- Le nombre d'IOPS par défaut pour les volumes racines est de 3 000. Vous pouvez définir un nombre maximal de 16 000 IOPS.

- Le débit du volume racine par défaut est de 125 Mbits/s. Vous pouvez définir un débit maximal de 1 000 MiB/s.

> **Note**
>
> Le rapport taille du volume racine/IOPS ne peut pas être supérieur à 1 volume pour 500 IOPS (1:500), et le rapport IOPS/débit du volume racine ne peut pas être supérieur à 1 IOPS pour un débit de 0,25 (1:0,25).

Pour plus d'informations sur Amazon EBS, consultez la section Volume de périphérique racine Amazon EC2.

Type de volume du périphérique racine avec l'AMI par défaut

Lorsque vous utilisez l'AMI par défaut, le type de volume du périphérique racine est déterminé par la version Amazon EMR que vous utilisez.

- À partir de la version 6.15.0, Amazon EMR attache un SSD à usage général (gp3) comme type de volume du périphérique racine.
- Pour les versions antérieures, Amazon EMR attache un SSD à usage général (gp2) comme type de volume du périphérique racine.

Type de volume du périphérique racine avec l'AMI personnalisée

Une AMI personnalisée peut avoir différents types de volumes de périphérique racine. Amazon EMR utilise toujours le type de volume de votre AMI personnalisée.

- Avec Amazon EMR 6.15.0 et versions ultérieures, vous pouvez configurer la taille du volume racine, les IOPS et le débit pour votre AMI personnalisée, à condition que ces attributs correspondent au type de volume de votre AMI personnalisée.
- Pour les versions antérieures, vous pouvez uniquement configurer la taille du volume racine pour votre AMI personnalisée.

Si vous ne configurez pas la taille du volume racine, les IOPS ou le débit lors de la création de votre cluster, Amazon EMR utilise les valeurs de l'AMI personnalisée, le cas échéant. Si vous décidez de configurer ces valeurs lors de la création de votre cluster, Amazon EMR utilise les valeurs que vous

spécifiez, à condition qu'elles soient compatibles et prises en charge par le volume racine de l'AMI personnalisée. Pour plus d'informations, consultez Utilisation d'une image AMI personnalisée.

Tarification relative à la taille du volume du périphérique racine

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 facturés au nombre de Go, mais vous spécifiez la taille du volume racine en Gio. Tenez-en compte dans vos estimations (1 Go = 0,931323 Gio).

Les SSD à usage général gp2 et gp3 sont facturés différemment. Pour estimer les frais associés aux volumes du périphérique racine EBS de votre cluster, utilisez la formule suivante :

SSD à usage général gp2

Le coût de gp2 dépend uniquement de la taille du volume EBS en Go.

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *
 InstanceCount
```

Par exemple, prenez un cluster doté d'un nœud primaire, d'un nœud principal et 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).

SSD à usage général gp3

Le coût de gp3 dépend de la taille du volume EBS en Go, du nombre d'IOPS au-delà de 3 000 IOPS (3 000 IOPS inclus) et le débit supérieur à 125 Mo/s (125 Mo/s inclus).

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *
 InstanceCount
+
($EBS IOPS/Month)/30/24* (EMR_EBSRootVolumeIops - 3000) * InstanceCount
+
($EBS throughput/Month)/30/24* (EMR_EBSRootVolumeThroughputInMb/s - 125) *
 InstanceCount
```

Par exemple, prenez un cluster doté d'un nœud primaire, d'un nœud principal et qui utilise l'AMI Amazon Linux de base avec le volume du périphérique racine de 15 Gio par défaut, 4 000 IOPS et un débit de 140 Mo/s. Si le coût de l'EBS dans la région est de 0,10 USD/Go/mois, le calcul est le suivant : 0,005 USD/IOPS provisionnés/mois au-delà de 3 000 et 0,040 USD/Mo/mois au-delà de 125 Mo/s. Cela correspond à environ 0,009293 USD par instance et par heure, et à 0,018586 USD par heure pour le cluster.

Personnalisation des paramètres du volume du périphérique racine

> **Note**
>
> Le rapport taille du volume racine/IOPS ne peut pas être supérieur à 1 volume pour 500 IOPS (1:500), et le rapport IOPS/débit du volume racine ne peut pas être supérieur à 1 IOPS pour un débit de 0,25 (1:0,25).

Console

Pour spécifier les attributs du volume du périphérique racine Amazon EBS dans la console Amazon EMR

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sélectionnez Amazon EMR 6.15.0 ou une version ultérieure.
4. Sous Configuration du cluster, accédez à la section Volume racine EBS et entrez une valeur pour les attributs que vous souhaitez configurer.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

CLI

Pour spécifier la taille du volume du périphérique racine Amazon EBS dans la AWS CLI

- Utilisez les paramètres `--ebs-root-volume-size`, `--ebs-root-volume-iops` et `--ebs-root-volume-throughput` de la commande create-cluster, comme illustré dans l'exemple suivant.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --release-label emr-6.15.0\
--ebs-root-volume-size 20 \
--ebs-root-volume-iops 3000\
--ebs-root-volume-throughput 135\
--instance-groups InstanceGroupType=MASTER,\
InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge
```

Configuration des logiciels de cluster

Lorsque vous sélectionnez une version logicielle, Amazon EMR utilise une Amazon Machine Image (AMI) avec Amazon Linux pour installer le logiciel que vous choisissez lors du lancement de votre cluster, tel que Hadoop, Spark et Hive. Amazon EMR fournit régulièrement de nouvelles versions, ajoutant 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 chaque version, consultez le Guide de version Amazon EMR. Pour plus d'informations sur la manière de modifier les configurations par défaut des applications et des logiciels installés sur votre cluster, accédez à la section Configuration des applications dans le Guide de mise à jour Amazon EMR. Certaines versions des composants de l'écosystème open source Hadoop et Spark qui sont

incluses dans les versions Amazon EMR comportent des correctifs et des améliorations, qui sont documentés dans le Guide de mise à jour Amazon EMR.

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 Bootstrap sont également utiles pour appeler des AWS CLI commandes sur chaque nœud afin de copier des objets depuis Amazon S3 vers 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 2.x et 3.x de l'AMI Amazon EMR, consultez la section Différences introduites dans les versions 4.x dans le Guide de version Amazon EMR.

Création d'actions d'amorçage pour installer des logiciels supplémentaires

Vous pouvez utiliser une action d'amorçage pour installer un logiciel supplémentaire 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'Amazon Machine Image (AMI) Amazon 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 d'Amazon EMR. Par exemple, `configure-Hadoop` et `configure-daemons` ne sont pas pris en charge dans la version 4.x d'Amazon EMR. La version 4.x d'Amazon EMR propose plutôt cette fonctionnalité à l'origine. Pour plus d'informations sur la façon de migrer les actions d'amorçage des versions 2.x et 3.x de l'AMI Amazon EMR vers la version 4.x d'Amazon EMR, consultez la section Personnalisation de la configuration des clusters et des applications avec les versions antérieures de l'AMI d'Amazon EMR dans le Guide de version Amazon EMR.

Principes de base de l'action d'amorçage

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 d'Amazon EMR prennent en charge les actions de démarrage. Vous pouvez spécifier jusqu'à 16 actions de bootstrap par cluster en fournissant plusieurs `bootstrap-actions` paramètres depuis la console ou AWS CLI 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`.

```
--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.

Exécuter une action d'amorçage de manière conditionnelle

Afin de n'exécuter une action d'amorçage que sur le nœud principal, vous pouvez utiliser une action d'amorçage personnalisée avec une certaine logique pour déterminer si le nœud est principal.

```
#!/bin/bash
if grep isMaster /mnt/var/lib/info/instance.json | grep false;
then
    echo "This is not master node, do nothing,exiting"
    exit 0
fi
echo "This is master, continuing to execute script"
# continue with code logic for master node below
```

La sortie suivante sera imprimée à partir d'un nœud principal.

```
This is not master node, do nothing, exiting
```

La sortie suivante sera imprimée à partir du nœud principal.

```
This is master, continuing to execute script
```

Pour utiliser cette logique, chargez votre action d'amorçage, y compris le code ci-dessus, dans votre compartiment Amazon S3. Sur le AWS CLI, ajoutez le `--bootstrap-actions` paramètre à l'appel d'`aws emr create-cluster`API et spécifiez l'emplacement de votre script bootstrap comme valeur de`Path`.

Actions de fin de tâche

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 primaire à l'aide de SSH.

Utilisation d'actions d'amorçage personnalisées

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.

> **Note**
>
> Pour de meilleures performances, nous vous recommandons de stocker les actions d'amorçage personnalisées, les scripts et les autres fichiers que vous souhaitez utiliser avec

Amazon EMR dans un compartiment Amazon S3 Région AWS identique à celui de votre cluster.

Table des matières

- Ajout d'actions d'amorçage personnalisées
- Utilisation d'une action d'amorçage personnalisée pour copier un objet depuis Amazon S3 vers chaque nœud

Ajout d'actions d'amorçage personnalisées

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour créer un cluster avec une action d'amorçage à l'aide de la nouvelle console.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Actions d'amorçage, choisissez Ajouter pour spécifier un nom, l'emplacement du script et des arguments facultatifs pour votre action. Sélectionnez Ajouter une action d'amorçage.
4. En option, ajoutez d'autres actions d'amorçage.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour créer un cluster avec une action d'amorçage personnalisée avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster.
3. Cliquez sur Accéder aux options avancées.
4. Dans Créer un cluster - Options avancées, Étapes 1 et 2, choisissez les options que vous souhaitez, puis passez à Step 3: General Cluster Settings (Étape 3 : Paramètres généraux de 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 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.

   Tant que le nœud primaire du cluster est en cours d'exécution, vous pouvez vous connecter au nœud primaire et voir les fichiers journaux que le script d'action d'amorçage a généré dans le répertoire `/mnt/var/log/bootstrap-actions/1`.

CLI

Pour créer un cluster avec une action bootstrap personnalisée à l'aide du AWS CLI

Lorsque vous utilisez l'action AWS CLI pour inclure un bootstrap, spécifiez le `Path` et `Args` sous forme de 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, saisissez la commande suivante, en remplaçant *myKey* par le nom de votre paire de clés EC2. Incluez `--bootstrap-actions` en tant que paramètre et spécifiez l'emplacement de votre script d'amorçage sous la forme de `Path`.

  - Utilisateurs Linux, UNIX et Mac OS X :

```
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"
```

- Utilisateurs Windows :

```
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 primaire 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Utilisation d'une action d'amorçage personnalisée pour copier un objet depuis Amazon S3 vers chaque nœud

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. AWS CLI Il est installé sur chaque nœud d'un cluster, de sorte que votre action bootstrap peut appeler des AWS CLI commandes.

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 sous le nom de fichier `copymyfile.sh` avec le contenu suivant.

```
#!/bin/bash
```

```
aws s3 cp s3://mybucket/myfilefolder/myfile.jar /mnt1/myfolder
```

Lorsque vous lancez le cluster, vous spécifiez le script. L' AWS CLI exemple suivant illustre cela :

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.1.0 \
--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 Amazon EMR, il est important de tenir compte de la manière dont vous configurez les instances Amazon EC2 et les options réseau. Ce chapitre couvre les options suivantes, puis les relie toutes avec des bonnes pratiques et des directives.

- Types de nœuds – Les instances Amazon EC2 d'un cluster EMR sont organisées en types de nœuds. Il en existe trois : les nœuds primaires, les nœuds principaux et les nœuds de tâches. 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 de tâches traitent les données, transfèrent les résultats vers Amazon S3 ou HDFS et fournissent des métadonnées d'état au nœud principal. Dans le cas d'un cluster à un seul nœud, tous les composants s'exécutent sur le nœud primaire. Pour plus d’informations, consultez Comprendre les types de nœuds : nœuds principaux, principaux et de tâches.
- Instances EC2 : lorsque vous créez un cluster, vous faites des choix concernant les instances Amazon EC2 sur lesquelles chaque type de nœud sera exécuté. Le type d'instance EC2 détermine le profil de traitement et de stockage du nœud. Le choix de l'instance Amazon EC2 pour vos nœuds est important car il détermine le profil de performance des différents types de nœuds de votre cluster. Pour plus d’informations, consultez Configuration des instances Amazon EC2.
- Mise en réseau : vous pouvez lancer votre cluster Amazon EMR dans un VPC à l'aide d'un sous-réseau public, d'un sous-réseau privé ou d'un sous-réseau partagé. Votre configuration réseau détermine la manière dont les clients et les services peuvent se connecter aux clusters pour effectuer des tâches, la manière dont les clusters se connectent aux magasins de données et aux autres ressources AWS , ainsi que les options dont vous disposez pour contrôler le trafic sur ces connexions. Pour plus d’informations, consultez Configuration de la mise en réseau.

- Groupement d'instances – La collection d'instances EC2 qui hébergent chaque type de nœud est appelée soit un parc d'instances, soit un groupe d'instances uniforme. La configuration du groupement d'instances est un choix que vous faites lorsque vous créez un cluster. Ce choix détermine la manière dont vous pouvez ajouter des nœuds à votre cluster pendant son exécution. La configuration s'applique à tous les types de nœuds. Il ne peut pas être modifié ultérieurement. Pour plus d'informations, consultez Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes.

  > **Note**
  >
  > La configuration de flotte 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.

Comprendre les types de nœuds : nœuds principaux, principaux et de tâches

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 primaire

Le nœud primaire gère le cluster et exécute généralement les composants primaires des applications distribuées. Par exemple, le nœud principal exécute le ResourceManager service YARN pour gérer les ressources des applications. Il exécute également le NameNode service HDFS, suit l'état des tâches soumises au cluster et surveille l'état des groupes d'instances.

Pour surveiller la progression d'un cluster et interagir directement avec les applications, vous pouvez vous connecter au nœud primaire via SSH en tant qu'utilisateur Hadoop. Pour plus d'informations, consultez Connexion au nœud primaire à l'aide de SSH. La connexion au nœud primaire vous permet d'accéder directement aux répertoires et aux fichiers, tels que les fichiers journaux Hadoop. Pour plus d'informations, consultez Afficher les fichiers journaux . Vous pouvez aussi afficher les interfaces utilisateur que les applications publient sous forme de sites web s'exécutant sur le nœud primaire. Pour plus d'informations, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

> **Note**
>
> Avec Amazon EMR 5.23.0 et versions ultérieures, vous pouvez lancer un cluster avec trois nœuds principaux pour prendre en charge la haute disponibilité d'applications telles que YARN Resource Manager, HDFS, Spark NameNode, Hive et Ganglia. Le nœud primaire n'est plus un point de défaillance potentiel grâce à cette fonctionnalité. Si l'un des nœuds primaires tombe en panne, Amazon EMR passe automatiquement sur un nœud primaire de secours et remplace le nœud primaire défaillant par un nouveau nœud ayant la même configuration et les mêmes actions de démarrage. Pour plus d'informations, consultez Planification et configuration des nœuds primaires.

Nœuds principaux

Les nœuds principaux sont gérés par le nœud primaire. 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 des NodeManager démons YARN, des MapReduce tâches Hadoop et des exécuteurs Spark.

Il n'existe qu'un seul groupe d'instances principal ou un seul parc d'instances par cluster, mais plusieurs nœuds peuvent s'exécuter sur plusieurs instances Amazon EC2 dans le groupe d'instances ou le parc d'instances. Avec les groupes d'instances, vous pouvez ajouter et supprimer des instances Amazon EC2 pendant que le cluster est en cours d'exécution. Vous pouvez également configurer le dimensionnement automatique pour ajouter des instances en fonction de la valeur d'une métrique. Pour plus d'informations sur l'ajout et le retrait d'instances Amazon EC2 avec la configuration des groupes d'instance, consultez Utiliser la mise à l'échelle des clusters.

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.

> **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 Quand faut-il utiliser des instances Spot ?.

Nœuds de tâches

Vous pouvez utiliser les nœuds de tâches pour augmenter la puissance nécessaire à l'exécution de tâches de calcul parallèles sur les données, telles que les tâches Hadoop et les exécuteurs MapReduce 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 Amazon EC2 à un groupe d'instances existant ou en modifiant les capacités cibles d'un parc d'instances de tâches.

Avec la configuration de groupe d'instances uniforme, vous pouvez avoir jusqu'à 48 groupes d'instances de tâches au total. La possibilité d'ajouter des groupes d'instances de cette manière vous permet de combiner des types d'instance Amazon EC2 et 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.

Avec la configuration de parc d'instances, la possibilité de mélanger les types d'instances et les 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 tâches, Amazon EMR dispose d'une fonctionnalité par défaut pour planifier les tâches YARN afin que les tâches en cours n'échouent pas lorsque les nœuds de tâches s'exécutant sur des instances Spot sont résiliés. Pour ce faire, Amazon EMR autorise les processus principaux de l'application à s'exécuter uniquement sur les nœuds principaux. Le processus principal de l'application contrôle les tâches en cours d'exécution et doit rester actif pendant toute la durée de vie de la tâche.

Les versions 5.19.0 et ultérieures d'Amazon EMR utilisent la fonctionnalité intégrée d'étiquettes de nœuds YARN 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 afin que le planificateur de capacité YARN et le planificateur équitable tirent parti des étiquettes des nœuds. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de manière à ce que les maîtres d'applications soient planifiés uniquement sur les nœuds portant le label CORE. La modification manuelle des propriétés associées dans les classifications de configuration de yarn-site et de capacity-scheduler, ou directement dans les fichiers XML associés, pourrait interrompre 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.

Configuration des instances Amazon EC2

Les instances EC2 se déclinent dans différentes configurations, appelées types d'instance. Les types d'instance ont des capacités de processeur, d'entrée/sortie et de stockage différentes. En plus du type d'instance, vous pouvez choisir différentes options d'achat pour les instances Amazon EC2. Vous pouvez spécifier différents types d'instance et options d'achat au sein des groupes d'instances uniformes ou des parcs d'instances. Pour plus d'informations, consultez Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes. Pour obtenir des conseils sur le choix des types d'instances et des options d'achat pour votre application, consultez Bonnes pratiques pour la configuration des clusters.

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide du AWS Management Console, le nombre de vCPU indiqué pour chaque type d'instance est le nombre de vcores YARN pour ce type d'instance, et non le nombre de vCPU EC2 pour ce type d'instance. Pour plus d'informations sur le nombre de vCPU pour chaque type d'instance, consultez Types d'instances Amazon EC2.

Rubriques

- Types d’instance pris en charge
- Configuration de la mise en réseau
- Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes

Types d’instance pris en charge

Cette section décrit les types d'instances pris en charge par Amazon EMR, organisés par Région AWS. Pour en savoir plus sur les types d'instances, consultez Instances Amazon EC2 et Matrice des types d'instances d'AMI Amazon Linux.

Tous les types d'instances ne sont pas disponibles dans toutes les régions, et la disponibilité des instances dépend de la disponibilité et de la demande dans la région et la zone de disponibilité spécifiées. La zone de disponibilité d'une instance est déterminée par le sous-réseau que vous utilisez pour lancer votre cluster.

Considérations

Tenez compte des points suivants lorsque vous choisissez des types d'instance pour votre cluster Amazon EMR.

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide du AWS Management Console, le nombre de vCPU indiqué pour chaque type d'instance est le nombre de vcores YARN pour ce type d'instance, et non le nombre de vCPU EC2 pour ce type d'instance. Pour plus d'informations sur le nombre de vCPU pour chaque type d'instance, consultez Types d'instances Amazon EC2.

- Si vous créez un cluster en utilisant un type d'instance qui n'est pas disponible dans la région et la zone de disponibilité spécifiées, votre cluster risque de ne pas être alloué ou d'être bloqué lors de l'allocation. Pour plus d'informations sur la disponibilité des instances, consultez la page de tarification d'Amazon EMR ou consultez les tableaux Types d'instances pris en charge par Région AWS de cette page.
- 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. Pour plus d'informations, consultez 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.
- Les pilotes NVIDIA et CUDA sont installés par défaut sur les types d'instance GPU.

Types d'instances pris en charge par Région AWS

Les tableaux suivants répertorient les types d'instances Amazon EC2 pris en charge par Amazon EMR, organisés par. Région AWS Les tableaux répertorient également les premières versions d'Amazon EMR des séries 5.x, 6.x et 7.x qui prennent en charge chaque type d'instance.

USA Est (Virginie du Nord) - us-east-1

Classe d'instance	Type d'instance	Version minimale d'Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.3.0
	g6.2xlarge	emr-7.3.0
	g6.4xlarge	emr-7.3.0
	g6.8xlarge	emr-7.3.0
	g6.12xlarge	emr-7.3.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g6.16xlarge	emr-7.3.0
	g6.24xlarge	emr-7.3.0
	g6.48xlarge	emr-7.3.0
	gr6.4xlarge	emr-7.3.0
	gr6.8xlarge	emr-7.3.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

USA Est (Ohio) - us-east-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.3.0
	g6.2xlarge	emr-7.3.0
	g6.4xlarge	emr-7.3.0
	g6.8xlarge	emr-7.3.0
	g6.12xlarge	emr-7.3.0
	g6.16xlarge	emr-7.3.0
	g6.24xlarge	emr-7.3.0
	g6.48xlarge	emr-7.3.0
	gr6.4xlarge	emr-7.3.0
	gr6.8xlarge	emr-7.3.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

USA Ouest (Californie du Nord) – us-west-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

USA Ouest (Oregon) - us-west-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.3.0
	g6.2xlarge	emr-7.3.0
	g6.4xlarge	emr-7.3.0
	g6.8xlarge	emr-7.3.0
	g6.12xlarge	emr-7.3.0
	g6.16xlarge	emr-7.3.0
	g6.24xlarge	emr-7.3.0
	g6.48xlarge	emr-7.3.0
	gr6.4xlarge	emr-7.3.0
	gr6.8xlarge	emr-7.3.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

AWS GovCloud (US-Ouest) - -1 us-gov-west

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

AWS GovCloud (USA Est) - -1 us-gov-east

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Afrique (Le Cap) – af-south-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Mémoire optimisée	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Stockage optimisé	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Hong Kong) – ap-east-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Jakarta) – ap-southeast-3

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m6g.xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.2xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.4xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.8xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.12xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul accéléré	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r7i.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.48xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.3xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Mumbai) – ap-south-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Région Asie-Pacifique (Hyderabad) – ap-south-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Mémoire optimisée	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Osaka) – ap-northeast-3

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Séoul) – ap-northeast-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asie-Pacifique (Singapour) – ap-southeast-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asie-Pacifique (Sydney) - ap-southeast-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.32xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.48xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asie-Pacifique (Tokyo) - ap-northeast-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.32xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.48xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Canada (Centre) – ca-central-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i-flex.2xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Canada Ouest (Calgary) – ca-west-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.9xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3en.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Chine (Ningxia) – cn-northwest-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Chine (Beijing) – cn-north-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europe (Francfort) eu-central-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (Zurich) – eu-central-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europe (Irlande) – eu-west-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (Londres) – eu-west-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
Calcul accéléré	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Stockage optimisé	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (Milan) – eu-south-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europe (Espagne) – eu-south-2

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Europe (Paris) – eu-west-3

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (Stockholm) – eu-north-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7a.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7a.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Moyen-Orient (Bahreïn) – me-south-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Calcul optimisé	c5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Moyen-Orient (Émirats arabes unis) – me-central-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Calcul accéléré	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Stockage optimisé	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Amérique du Sud (São Paulo) – sa-east-1

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
Usage général	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul optimisé	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
Calcul accéléré	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Mémoire optimisée	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Stockage optimisé	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d’instance	Version minimale d’Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Classe d'instance	Type d'instance	Version minimale d'Amazon EMR prise en charge (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

instances de la génération précédente

Amazon EMR prend en charge les Instances de la génération précédente 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'instances
General Purpose	m1.small[1] | m1.medium[1] | m1.large[1] | m1.xlarge[1] | m3.xlarge[1] | m3.2xlarge[1] | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
Compute Optimized	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
Memory Optimized	m2.xlarge[1] | m2.2xlarge[1] | m2.4xlarge[1] | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Storage Optimized	d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge

[1] Utilise l'AMI de virtualisation PVM avec les versions d'Amazon EMR antérieures à 5.13.0. Pour plus d'informations, consultez la page Types de virtualisations AMI Linux.

[2] Non pris en charge dans la version 5.15.0.

Options d’achat d’instance

Lorsque vous configurez un cluster, vous choisissez une option d'achat pour les instances Amazon EC2. Vous pouvez choisir les instances À la demande et/ou les instances Spot. Les prix varient en fonction du type d'instance et de la région. Le prix Amazon EMR s'ajoute au prix Amazon EC2 (le prix pour les serveurs sous-jacents) et au prix Amazon EBS (si vous joignez des volumes Amazon EBS). 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, vous ne pouvez spécifier l'option d'achat pour un groupe d'instances que lorsque vous le créez. Le type d'instance et l'option d'achat s'appliquent à toutes les instances Amazon EC2 de chaque groupe d'instances. 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.

On-Demand instances

Les instances à la demande vous permettent de payer la capacité de calcul à la seconde. 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 isolées physiquement au niveau du matériel hôte des instances appartenant à d'autres AWS comptes. Pour plus d'informations sur les options d'achat, consultez la section Options d'achat d'instances dans le guide de l'utilisateur Amazon EC2.

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 plus d'informations, consultez les sections Instances réservées Amazon EC2 et Achat d'instances réservées dans le Guide de l'utilisateur Amazon EC2. 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 Amazon 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.

Spot instances

Les instances Spot dans Amazon EMR vous permettent d'acheter de la capacité d'instance Amazon EC2 à un coût réduit par rapport à l'achat à la demande. L'inconvénient de l'utilisation des instances Spot est que les instances peuvent être résiliées si la capacité Spot devient indisponible pour le type d'instance que vous exécutez. Pour plus d'informations sur l'utilisation appropriée des instances Spot pour votre application, consultez Quand faut-il utiliser des instances Spot ?.

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.

> **Note**
>
> Les instances Spot de durée définie (également appelées blocs d'instances Spot) ne sont plus disponibles pour les nouveaux clients depuis le 1er juillet 2021. Pour les clients qui ont déjà utilisé cette fonctionnalité, nous continuerons à prendre en charge les instances Spot de durée définie jusqu'au 31 décembre 2022.

Pour connaître la tarification actuelle, consultez Tarification des instances Amazon EC2 Spot. Pour plus d'informations, consultez la section Instances Spot dans le guide de l'utilisateur Amazon EC2. 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 Configuration de la mise en réseau.

> **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 l'historique des tarifs des instances Spot dans le guide de l'utilisateur Amazon EC2.

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 une seule 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 Création d'un cluster avec des parcs d'instances ou des groupes d'instances uniformes.

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 est prise en compte dans votre capacité cible lorsqu'une instance de ce type est provisionnée. Amazon EMR approvisionne les instances avec les deux options d'achat jusqu'à ce que la capacité cible pour chaque objectif soit 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 fournissez également des options Spot supplémentaires pour chaque parc d'instances Spot, y compris un délai de provisionnement. Pour plus d'informations, consultez Configuration de parcs d'instances.

Stockage d'instances

Présentation

Le stockage d'instances et de volumes Amazon EBS est utilisé pour les données HDFS et pour les tampons, les caches, les données scratch et d'autres contenus temporaires que certaines applications peuvent « déborder » sur le système de fichiers local.

Amazon EBS fonctionne différemment au sein d'Amazon EMR par rapport aux instances Amazon EC2 classiques. Les volumes Amazon EBS attachés aux clusters Amazon EMR sont éphémères : ils sont supprimés à l'arrêt du cluster et de l'instance (par exemple, lors de la réduction des groupes d'instances). Ne vous attendez donc pas à ce que les données soient conservées. 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 Amazon EBS 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).

Considérations

Tenez compte des éléments supplémentaires suivants lorsque vous utilisez Amazon EBS avec des clusters EMR :

- Vous ne pouvez pas prendre un instantané d'un volume Amazon 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 Utilisation d'une image AMI personnalisée.
- Un volume de périphérique racine Amazon EBS chiffré n'est pris en charge que lors de l'utilisation d'une AMI personnalisée. Pour plus d'informations, consultez Création d'une AMI personnalisée avec un volume de périphérique racine Amazon EBS chiffré.
- 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 Amazon EBS sur les nœuds principaux ne peuvent pas être inférieurs à 5 Go.

Stockage Amazon EBS par défaut pour les instances

Pour les instances EC2 disposant d'un stockage exclusivement EBS, Amazon EMR alloue des volumes de stockage Amazon EBS gp2 ou gp3 aux instances. Lorsque vous créez un cluster avec Amazon EMR 5.22.0 et versions ultérieures, le volume de stockage Amazon EBS par défaut augmente en fonction de la taille de l'instance.

Les augmentations de stockage sont fractionnées sur plusieurs volumes. Cela permet d'augmenter les performances IOPS et, par conséquent, les performances de certaines charges de travail standardisées. Si vous souhaitez utiliser une configuration de stockage d'instance Amazon EBS différente, vous pouvez le spécifier lorsque vous créez un cluster EMR ou lorsque vous ajoutez des nœuds à un cluster existant. Vous pouvez uniquement utiliser les volumes Amazon EBS gp2 ou gp3 comme volumes racines et ajouter des volumes gp2 ou gp3 comme volumes supplémentaires. Pour plus d'informations, consultez Spécification de volumes de stockage EBS supplémentaires.

Le tableau suivant indique le nombre par défaut de volumes de stockage Amazon EBS gp2, les tailles et les tailles totales par type d'instance. Pour plus d'informations sur les différences entre les volumes gp2 et gp3, voir la rubrique Comparaison des types de volumes Amazon EBS gp2 et gp3.

Volumes de stockage Amazon EBS gp2 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

Volume racine Amazon EBS par défaut pour les instances

À partir de la version 6.15, Amazon EMR attache automatiquement un SSD à usage général Amazon EBS (gp3) comme périphérique racine pour ses AMI afin d'améliorer les performances. Dans les versions antérieures, Amazon EMR attache le SSD à usage général EBS (gp2) comme périphérique racine.

	6.15 et versions ultérieures	6.14 et versions antérieures
Type de volume racine par défaut		
Taille par défaut		
IOPS par défaut		
Débit par défaut		

Pour plus d'informations sur la personnalisation du volume du périphérique racine Amazon EBS, voir la rubrique Spécification de volumes de stockage EBS supplémentaires.

Spécification de volumes de stockage EBS supplémentaires

Lorsque vous configurez des types d'instances dans Amazon EMR, vous pouvez spécifier des volumes EBS supplémentaires pour ajouter de la capacité au-delà du stockage d'instances (le cas échéant) et du volume EBS par défaut. Amazon EBS fournit les types de volumes suivants : à usage général (SSD), IOPS provisionnés (SSD), optimisé pour le débit (HDD), à froid (HDD) et magnétique. Ils se distinguent par leurs performances et leur prix, ce qui vous permet d'adapter votre stockage en fonction des besoins opérationnels et d'analyse de vos applications. Par exemple, certaines applications peuvent avoir besoin de « déborder » sur le disque, tandis que d'autres peuvent travailler en toute sécurité dans la mémoire ou à l'aide d'Amazon S3.

Vous ne pouvez attacher des volumes Amazon EBS aux instances qu'au moment du démarrage du cluster et lorsque vous ajoutez un groupe d'instances de nœuds de tâches supplémentaires. Si une instance d'un cluster Amazon EMR échoue, l'instance et les volumes Amazon EBS attachés sont remplacés par de nouveaux volumes. Par conséquent, si vous détachez manuellement un volume Amazon EBS, Amazon EMR traite cela comme une défaillance et remplace le stockage d'instance (le cas échéant) et les stockages de volume.

Amazon EMR ne vous permet pas de modifier le type de volume de gp2 à gp3 pour un cluster EMR existant. Pour utiliser un volume gp3 pour vos charges de travail, lancez un nouveau cluster EMR. En outre, nous vous déconseillons de mettre à jour le débit et les IOPS sur un cluster en cours d'utilisation ou de provisionnement, car Amazon EMR utilise les valeurs de débit et d'IOPS que vous

avez spécifiées au moment du lancement du cluster pour toute nouvelle instance ajoutée lors de la mise à l'échelle du cluster. Pour plus d'informations, consultez Comparaison des types de volumes Amazon EBS gp2 et gp3 et Sélection des IOPS et du débit lors de la migration vers gp3.

> **Important**
>
> Pour utiliser un volume gp3 avec votre cluster EMR, vous devez lancer un nouveau cluster.

Comparaison des types de volumes Amazon EBS gp2 et gp3

Voici une comparaison des coûts entre les volumes gp2 et gp3 dans la région USA Est (Virginie du Nord). Pour obtenir les informations les plus récentes, consultez la page produit Volumes à usage général Amazon EBS et la page de tarification d'Amazon EBS.

Type de volume	gp3	gp2
Taille du volume	1 Gio – 16 Tio	1 Gio – 16 Tio
IOPS par défaut/de référence	3000	3 IOPS/Go (minimum 100 IOPS) pour un maximum de 16 000 IOPS. Les volumes inférieurs à 1 Tio peuvent également atteindre 3 000 IOPS.
IOPS maximum/volume	16,000	16,000
Débit par défaut/de référence	125 Mo/s	La limite de débit est comprise entre 128 Mio/s et 250 Mio/s, selon la taille du volume.
Débit maximal/volume	1,000 Mio/s	250 Mio/s
Prix	0,08 USD/Go par mois, 3 000 IOPS gratuites et 0,005 USD/ mois d'IOPS provisionnées par mois pour plus de 3 000 ; 125 Mbits/s gratuits et 0,04 USD/	0,10 USD/Go par mois

Type de volume	gp3	gp2
	Mio par mois provisionné au-dessus de 125 Mo/s	

Sélection des IOPS et du débit lors de la migration vers gp3

Lorsque vous approvisionnez un volume gp2, vous devez déterminer la taille du volume afin d'obtenir les IOPS et le débit proportionnels. Avec gp3, vous n'avez pas besoin de fournir un volume plus important pour obtenir de meilleures performances. Vous pouvez choisir la taille et les performances souhaitées en fonction des besoins de l'application. La sélection de la bonne taille et des bons paramètres de performance (IOPS, débit) peut vous permettre de réduire les coûts au maximum, sans affecter les performances.

Voici un tableau pour vous aider à sélectionner les options de configuration de la GP3 :

Taille du volume	IOPS	Débit
1 à 170 Gio	3000	125 Mo/s
170 à 334 Gio	3000	125 Mbits/s si le type d'instance EC2 choisi prend en charge 125 Mo/s ou moins, utilisez une valeur supérieure en fonction de l'utilisation, 250 Mi/s* maximum.
334 à 1 000 Gio	3000	125 Mio/s si le type d'instance EC2 choisi prend en charge 125 Mio/s ou moins, Utiliser plus selon l'utilisation, maximum 250 Mio/s*.
1000+ Gio	Correspond au nombre d'IOPS gp2 (taille en Gio x 3) ou au nombre maximal d'IOPS piloté par le volume gp2 actuel	125 Mio/s si le type d'instance EC2 choisi prend en charge 125 Mio/s ou moins, Utiliser plus selon l'utilisation, maximum 250 Mio/s*.

* Gp3 a la capacité de fournir un débit allant jusqu'à 1000 MiB/s. Comme gp2 fournit un débit maximal de 250 Mo/s, vous n'aurez peut-être pas besoin de dépasser cette limite lorsque vous utilisez gp3.

Configuration de la mise en réseau

La plupart des clusters sont lancés sur un réseau virtuel à l'aide d'Amazon Virtual Private Cloud (Amazon VPC). Un VPC est un réseau virtuel isolé au sein de celui-ci AWS qui est logiquement isolé au sein de votre compte. AWS Vous pouvez configurer des aspects tels que les plages d'adresses IP privées, les sous-réseaux, les tables de routage et les passerelles réseau. Pour de plus amples informations, consultez le Guide de l'utilisateur Amazon VPC.

&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 sur un réseau privé, il peut s'avérer peu pratique ou indésirable de télécharger ces données AWS pour les importer dans Amazon EMR, soit en raison de la quantité de données à transférer, soit en raison de leur nature sensible. 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 Amazon EMR dans des sous-réseaux VPC publics et privés. Cela signifie que vous n'avez pas besoin de connexion Internet pour exécuter un cluster Amazon EMR ; toutefois, vous devrez peut-être configurer la traduction d'adresses réseau (NAT) et des passerelles

VPN pour accéder à des services ou à des ressources situés en dehors du VPC, par exemple sur un intranet d'entreprise ou sur des points de terminaison de service public tels que. AWS AWS Key Management Service

> **Important**
>
> Amazon EMR ne prend en charge le lancement de clusters dans des sous-réseaux privés qu'à partir de la version 4.2.

Pour plus d'informations sur la sécurité dans Amazon VPC, veuillez consulter le Guide de l'utilisateur Amazon VPC.

Rubriques

- Options d'Amazon VPC
- Configuration d'un VPC pour héberger des clusters
- Lancement de clusters dans un VPC
- Politique Amazon S3 minimale pour le sous-réseau privé
- Ressources supplémentaires pour en savoir plus sur les VPC

Options d'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

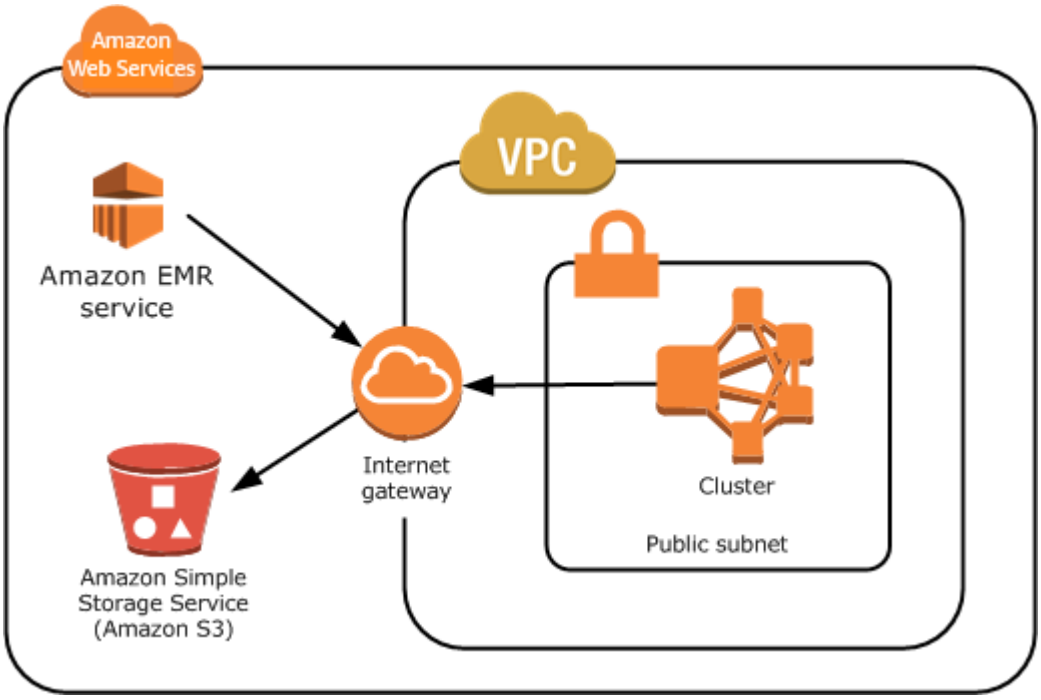
Les clusters EMR dans un sous-réseau public nécessitent une passerelle Internet connectée Cela est dû au fait que les clusters Amazon EMR doivent accéder aux AWS services et à Amazon EMR. Si un service, tel qu'Amazon S3, offre la possibilité de créer un point de terminaison d'un VPC, vous pouvez accéder 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 d'un VPC ou une passerelle Internet. AWS Les autres services qui ne prennent pas actuellement en charge les points de terminaison VPC utilisent uniquement une passerelle Internet.

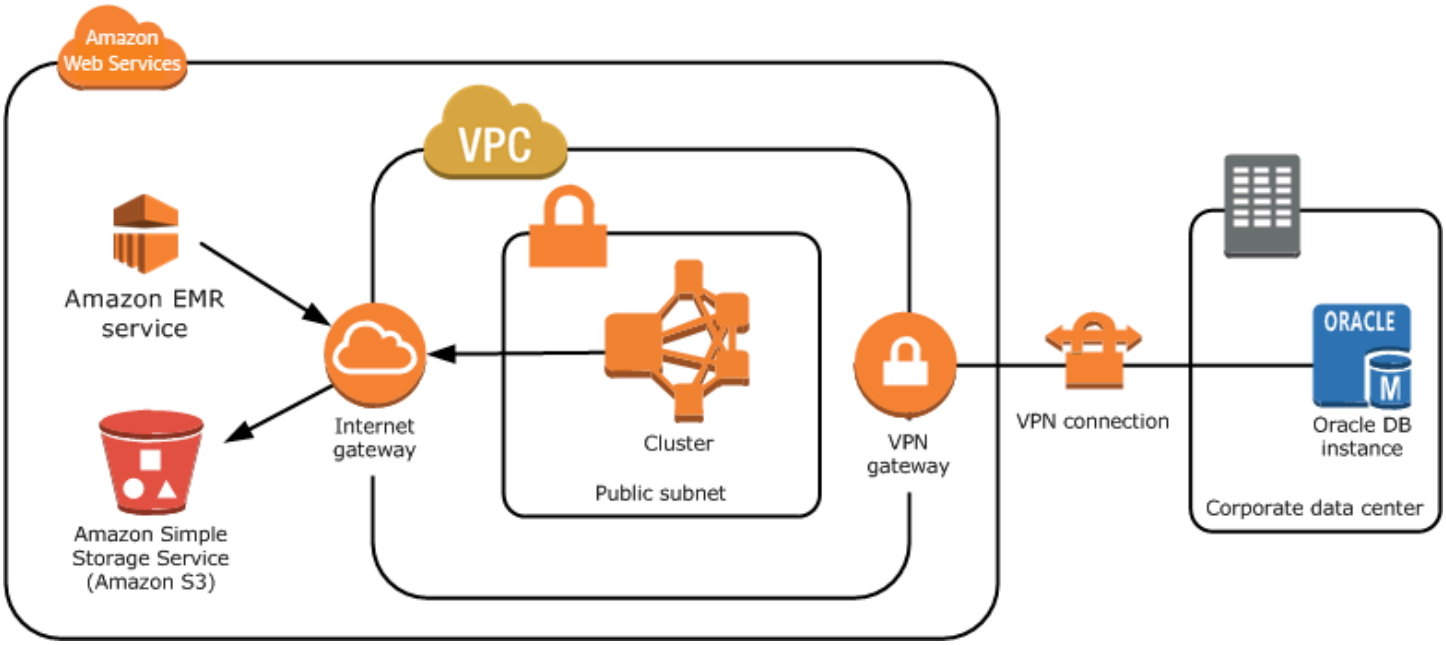
Si vous ne souhaitez pas connecter AWS des ressources supplémentaires à la passerelle Internet, vous pouvez lancer ces composants dans un sous-réseau privé que vous créez au sein de votre VPC.

Les clusters exécutant un sous-réseau public utilisent deux groupes de sécurité : un groupe pour le nœud primaire et un autre pour les nœuds de noyau et de tâche. Pour plus d'informations, consultez Contrôle du trafic réseau avec des groupes de sécurité.

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 AWS ressources, telles que les 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

Un sous-réseau privé vous permet de lancer AWS des ressources sans avoir besoin d'une passerelle Internet attachée au sous-réseau. Amazon EMR prend en charge le lancement de clusters dans des sous-réseaux privés avec les versions 4.2.0 ou ultérieures.

> **Note**
>
> Lorsque vous configurez un cluster Amazon EMR dans un sous-réseau privé, nous vous recommandons de configurer également des points de terminaison VPC pour Amazon S3. Si votre cluster EMR se trouve dans un sous-réseau privé sans points de terminaison VPC pour Amazon S3, vous devrez payer des frais de passerelle NAT supplémentaires associés au trafic S3, car le trafic entre votre cluster EMR et S3 ne restera pas dans votre VPC.

La différence entre les sous-réseaux privés diffère des sous-réseaux publics pour les raisons suivantes :

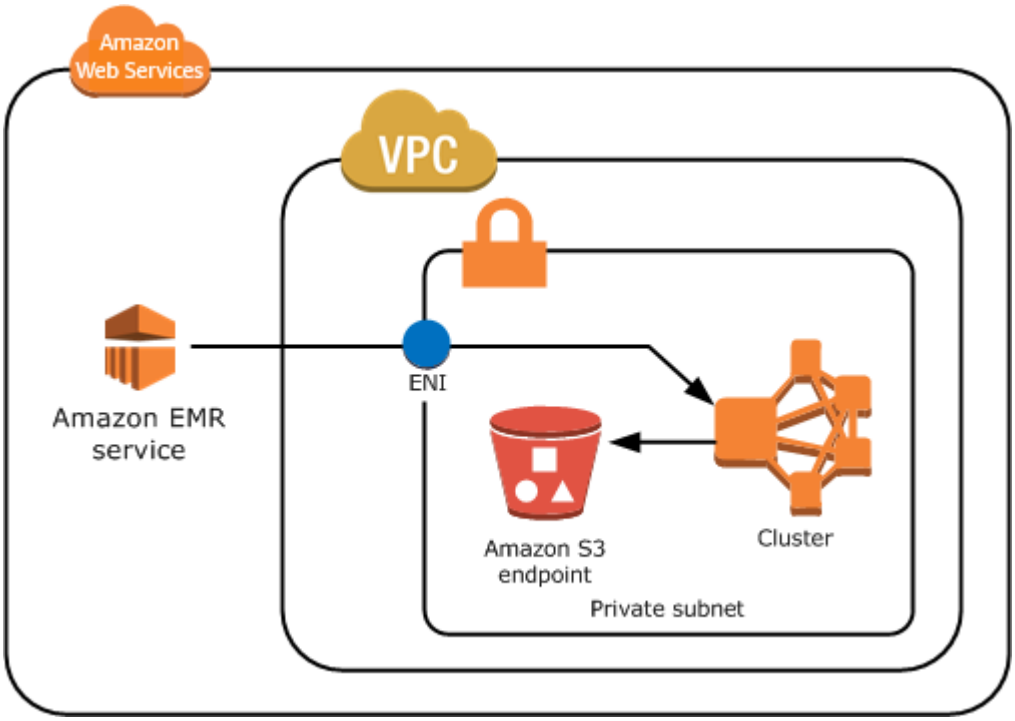
- Pour accéder aux AWS services qui ne fournissent pas de point de terminaison VPC, vous devez toujours utiliser une instance NAT ou une passerelle Internet.
- Au minimum, vous devez indiquer un chemin vers le compartiment des journaux du service Amazon EMR et vers le répertoire Amazon Linux dans Amazon S3. Pour plus d'informations, consultez Politique Amazon S3 minimale pour le sous-réseau privé.

- Si vous utilisez les fonctionnalités EMRFS, vous devez disposer d'un point de terminaison d'un VPC Amazon S3 et d'un acheminement 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 terminaison d'un VPC Amazon S3 pour les clusters Amazon EMR est d'utiliser la page Liste des sous-réseaux VPC dans la console Amazon EMR. Pour configurer les passerelles NAT, consultez la section Passerelles NAT dans le Guide de l'utilisateur Amazon VPC.
- Vous ne pouvez pas modifier un sous-réseau avec un cluster Amazon EMR existant de public à privé ou inversement. Pour placer un cluster Amazon EMR au sein d'un sous-réseau privé, le cluster doit être démarré dans ce sous-réseau privé.

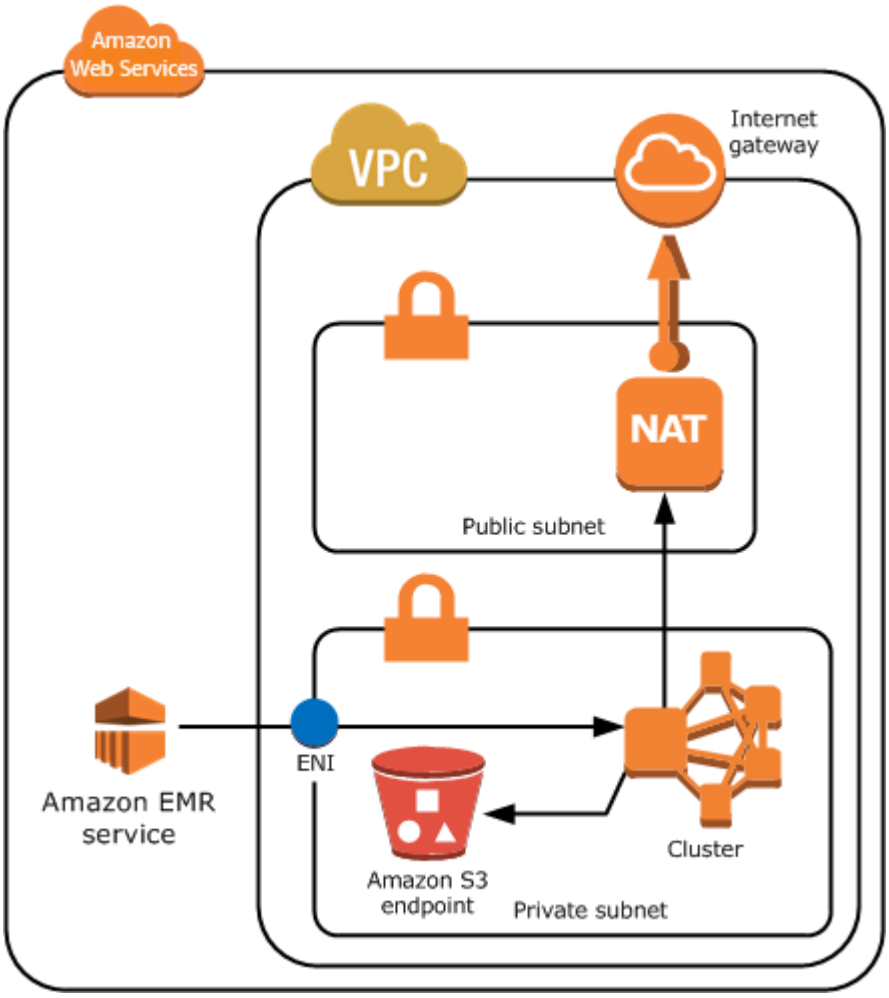
Amazon EMR crée et utilise différents groupes de sécurité par défaut pour les clusters d'un sous-réseau privé : ElasticMapReduce -Master-Private, Reduce-Slave-Private et -. ElasticMap ElasticMapReduce ServiceAccess Pour plus d’informations, consultez Contrôle du trafic réseau avec des groupes de sécurité.

Pour obtenir une liste complète des listes de contrôle d'accès réseau (listes ACL réseau) de votre cluster, choisissez Groupes de sécurité pour le principal et Groupes de sécurité pour le noyau et la tâche sur la page Détails du cluster de la console Amazon EMR.

L'image suivante montre comment un cluster Amazon 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 Amazon EMR au sein d'un sous-réseau privé connecté à une instance NAT située dans un sous-réseau public.



Sous-réseaux partagés

Le partage VPC permet aux clients de partager des sous-réseaux avec d'autres AWS comptes 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.

Contrôlez les autorisations VPC avec IAM

Par défaut, tous les utilisateurs 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 restreindre les actions à l'aide de politiques, comme vous le feriez avec les clusters lancés dans Amazon EC2 Classic. Pour plus d'informations sur IAM, consultez le Guide de l'utilisateur IAM.

Vous pouvez également utiliser IAM pour contrôler les personnes autorisées à créer et gérer des sous-réseaux. Par exemple, vous pouvez créer un compte pour administrer les sous-réseaux et un second compte qui peut lancer des clusters mais ne peut pas modifier les paramètres Amazon VPC. Pour plus d'informations sur l'administration des politiques et des actions dans Amazon EC2 et Amazon VPC, consultez la section Politiques IAM pour Amazon EC2 dans le guide de l'utilisateur Amazon EC2.

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 VPC avec des sous-réseaux pour un cluster Amazon EMR

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. En haut à droite de la page, choisissez la Région AWS pour votre VPC.
3. Sélectionnez Create VPC (Créer un VPC).
4. Sur la page Paramètres VPC, choisissez VPC et plus.
5. Sous Génération automatique des balises de nom, activez Générer automatiquement et entrez un nom pour votre VPC. Cela vous permet d'identifier le VPC et le sous-réseau dans la console Amazon VPC une fois que vous les avez créés.
6. Dans le champ Bloc d'adresse CIDR IPv4, entrez un espace d'adresses IP privé pour votre VPC afin de garantir une résolution correcte des noms d'hôte DNS ; dans le cas contraire, vous risquez de subir 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
7. Sous Number of Availability Zones (AZs) (Nombre de zones de disponibilité (AZ)), choisissez le nombre de zones de disponibilité dans lesquelles vous voulez lancer vos sous-réseaux.
8. Sous Nombre de sous-réseaux publics, choisissez un seul sous-réseau public à ajouter à votre VPC. Si les données utilisées par le cluster sont disponibles sur Internet (par exemple, dans Amazon S3 ou Amazon RDS), vous devez uniquement utiliser un sous-réseau public et vous n'avez pas besoin d'ajouter de sous-réseau privé.
9. Sous Number of private subnets (Nombre de sous-réseaux privés), choisissez le nombre de sous-réseaux privés que vous voulez ajouter à votre VPC. Sélectionnez-en une ou plusieurs si les données de votre application sont stockées sur votre propre réseau (par exemple, dans une base de données Oracle). Pour un VPC dans un sous-réseau privé, toutes les instances Amazon EC2 doivent avoir au moins un chemin vers Amazon EMR via l'interface réseau Elastic. Dans la console, ce paramètre est configuré automatiquement.
10. Sous Passerelles NAT, choisissez éventuellement d'ajouter des passerelles NAT. Ils ne sont nécessaires que si vous avez des sous-réseaux privés qui doivent communiquer avec Internet.
11. Sous Points de terminaison d'un VPC, choisissez éventuellement d'ajouter des points de terminaison pour Amazon S3 à vos sous-réseaux.
12. Vérifiez que les cases Activer les noms d'hôte DNS et Activer la résolution DNS sont cochées. Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.

13. Sélectionnez Create VPC (Créer un VPC).
14. Une fenêtre d'état indique la progression de ces tâches. Lorsque le travail est terminé, choisissez Afficher le VPC pour accéder à la page Vos VPC, qui affiche votre VPC par défaut et le VPC que vous venez de créer. Le VPC que vous avez créé étant un VPC personnalisé, la colonne Default VPC indique No.
15. Si vous souhaitez associer votre VPC à une entrée DNS qui n'inclut pas de nom de domaine, accédez aux Ensembles d'options DHCP, choisissez Créer un ensemble d'options DHCP et omettez un nom de domaine. Après avoir créé votre ensemble d'options, accédez à votre nouveau VPC, choisissez Modifier le jeu d'options DHCP dans le menu Actions, puis sélectionnez le nouveau jeu d'options. 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.

    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`**. Par exemple, dans `us-west-2`, utilisez **`us-west-2.compute.internal`**. Pour la région AWS GovCloud (ouest des États-Unis), utilisez**`us-gov-west-1.compute.internal`**.
    - domain-name-servers (serveurs-nom-domaine) = **`AmazonProvidedDNS`**

    Pour en savoir plus, consultez Jeux d'options DHCP dans le Guide de l'utilisateur Amazon VPC.
16. Une fois le VPC créé, accédez à la page Sous-réseaux et notez l'ID de sous-réseau de l'un des sous-réseaux de votre nouveau VPC. Vous utilisez ces informations lorsque vous lancez le cluster Amazon 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 dans les versions 4.2 et supérieures.

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'entrée sur le port 8443 pour communiquer avec le 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 devra peut-être ajouter des adresses IP supplémentaires à la AWS gamme au fil du temps. Pour plus d'informations, consultez Contrôle du trafic réseau avec des groupes de sécurité.

Pour gérer le cluster sur un VPC, Amazon EMR lie un périphérique réseau au nœud primaire et le gère par le biais de ce dispositif. Vous pouvez afficher ce dispositif à l'aide de l'action d'API Amazon EC2 `DescribeInstances`. Si vous modifiez ce dispositif, le cluster peut échouer.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour lancer un cluster dans un VPC avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Mise en réseau, accédez au champ Cloud privé virtuel (VPC). Entrez le nom de votre VPC ou choisissez Parcourir pour sélectionner votre VPC. Vous pouvez également choisir Créer un VPC pour créer un VPC que vous pouvez utiliser pour votre cluster.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour lancer un cluster dans un VPC avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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:RevokeSecurityGroupEgressec2:AuthorizeSecurityGroupEgress`,
> `cloudformation:DescribeStackEvents` et.
> `cloudformation:CreateStack`

> **Note**
>
> Des frais supplémentaires s'appliquent au lancement d'une instance Amazon EC2 pour votre périphérique NAT.

6. Procédez à la création du cluster.

AWS CLI

Pour lancer un cluster dans un VPC avec AWS CLI

> **Note**
>
> Il AWS CLI ne permet pas de créer automatiquement une instance NAT 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 Amazon 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* par le nom de votre paire de clés Amazon EC2 et remplacez *77XXXX03* par l'ID de votre 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 primaire 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`.

Politique Amazon S3 minimale pour le sous-réseau privé

Pour les sous-réseaux privés, vous devez au minimum permettre à Amazon EMR d'accéder aux référentiels Amazon Linux. Cette politique de sous-réseau privé fait partie des politiques du point de terminaison d'un VPC pour l'accès à Amazon S3. 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. Si vous activez la journalisation, accordez des autorisations PUT à un compartiment `aws157-logs-*`. 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 un nom de compartiment Amazon S3 ambigu. L'exemple de stratégie suivant fournit des autorisations pour accéder aux référentiels Amazon Linux et au compartiment système Amazon EMR pour la collecte des journaux d'événements Spark. *MyRegion*Remplacez-le par la région où se trouvent vos compartiments à journaux, par exemple`us-east-1`.

Pour plus d'informations sur l'utilisation des politiques IAM avec les points de terminaison d'un VPC Amazon, consultez Politiques de points de terminaison pour Amazon S3.

```
{
   "Version": "2008-10-17",
   "Statement": [
       {
           "Sid": "AmazonLinuxAMIRepositoryAccess",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": [
            "arn:aws:s3:::packages.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.emr.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/*"
            ]
        }
    ]
}
```

L'exemple de politique suivant fournit les autorisations requises pour accéder aux référentiels Amazon Linux 2. L'AMI Amazon Linux 2 est l'AMI par défaut.

```
{
    "Statement": [
        {
            "Sid": "AmazonLinux2AMIRepositoryAccess",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": [
             "arn:aws:s3:::amazonlinux.MyRegion.amazonaws.com/*",
             "arn:aws:s3:::amazonlinux-2-repos-MyRegion/*"
            ]
        }
    ]
}
```

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 des sous-réseaux publics et privés (NAT)
  - Instances NAT
  - Solution haute disponibilité des instances NAT sur 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 User Guide

- 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 primaire, 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.

Vous pouvez utiliser la console Amazon EMR AWS CLI, ou l'API Amazon EMR pour créer des clusters avec l'une ou l'autre configuration. 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 Amazon 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`.

Dans la nouvelle console Amazon EMR, vous pouvez choisir d'utiliser des groupes d'instances ou des parcs d'instances lorsque vous créez un cluster, et vous avez la possibilité d'utiliser des instances Spot avec chacun d'eux. Avec l'ancienne console Amazon EMR, si vous utilisez les paramètres Options rapides par défaut lorsque vous créez votre cluster, Amazon EMR applique la configuration de 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.

Flottes d'instances

La configuration des parcs d'instances offre la plus grande variété d'options de mise en service pour les instances Amazon EC2. Chaque type de nœud dispose d'un seule parc d'instances. L'utilisation d'un parc d'instances de tâches est facultative. Vous pouvez spécifier jusqu'à cinq types d'instances EC2 par parc, ou 30 types d'instances EC2 par parc lorsque vous créez un cluster à l'aide de l'API AWS CLI Amazon EMR et d'une stratégie d'allocation pour les instances à la demande et

ponctuelles. Pour les parcs d'instances principaux et de tâches, vous affectez une capacité cible pour les instances à la demande et une autre pour les instances Spot. Amazon EMR n'importe quelle combinaison des types d'instance spécifiés pour remplir les capacités cibles, en mettant en service des instances à la demande et des instances Spot.

Pour le type de nœud primaire, Amazon EMR choisit un seul type d'instance dans votre liste d'instances, et vous spécifiez s'il est alloué en tant qu'instance à la demande ou en tant qu'instance Spot. Les parcs d'instances proposent également des options supplémentaires pour les achats d'instances Spot et à la demande. Les options d'instance Spot incluent un délai d'expiration qui spécifie une action à entreprendre si la capacité ponctuelle ne peut pas être provisionnée, et une stratégie d'allocation préférée (optimisée pour les capacités) pour le lancement de parcs d'instances Spot. Les parcs d'instances à la demande peuvent également être lancées à l'aide de l'option de stratégie d'allocation (prix le plus bas). Si vous utilisez un rôle de service qui n'est pas le rôle de service EMR par défaut, ou si vous utilisez une politique gérée EMR dans votre rôle de service, vous devez ajouter des autorisations supplémentaires au rôle de service de cluster personnalisé pour activer l'option de stratégie d'allocation. Pour plus d’informations, consultez Rôle de service pour Amazon EMR (rôle EMR).

Pour plus d'informations sur la configuration des parcs d'instances, consultez Configuration de parcs d'instances.

Groupes d'instances uniformes

Les groupes d'instances uniformes offrent une configuration plus simple que les parcs d'instances. Chaque cluster Amazon EMR peut inclure jusqu'à 50 groupes d'instances : un groupe d'instances principales qui contient une instance Amazon EC2, un groupe d'instances principales 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 Amazon EC2. Vous pouvez dimensionner chaque groupe d'instances en ajoutant et en retirant des instances Amazon EC2 manuellement, ou vous pouvez définir un dimensionnement automatique. Pour plus d'informations sur l'ajout et le retrait d'instances, consultez Utiliser la mise à l'échelle des clusters.

Pour plus d'informations sur la configuration des groupes d'instances uniformes, consultez Configuration de groupes d'instances uniformes.

Utilisation de parcs d'instances et de groupes d'instances

Rubriques

- Configuration de parcs d'instances

- Utilisation des réserves de capacité avec les parcs d'instances
- Configuration de groupes d'instances uniformes
- Bonnes pratiques pour la flexibilité des instances et des zones de disponibilité
- Bonnes pratiques pour la configuration des clusters

Configuration de parcs d'instances

> **Note**
>
> La configuration de flotte 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.

La configuration du parc d'instances pour les clusters Amazon EMR vous permet de sélectionner une grande variété d'options de provisionnement pour les instances Amazon EC2 et vous aide à développer une stratégie de ressources flexible et élastique pour chaque type de nœud de votre cluster.

Dans une configuration de parc d'instances, vous spécifiez une capacité cible pour les instances à la demande et les instances Spot au sein de chaque parc. Lorsque le cluster se lance, Amazon EMR met en service des instances jusqu'à ce que les cibles soient atteintes. Lorsqu'Amazon EC2 récupère une instance Spot dans un cluster en cours d'exécution en raison d'une augmentation de prix ou d'une défaillance d'instance, Amazon EMR tente 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.

Vous pouvez spécifier un maximum de cinq types d'instances Amazon EC2 par parc pour qu'Amazon EMR puisse utiliser pour atteindre les objectifs, ou un maximum de 30 types d'instances Amazon EC2 par parc lorsque vous créez un cluster à l'aide de l'API AWS CLI Amazon EMR ou d'une stratégie d'allocation pour les instances ponctuelles et à la demande.

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. Si Amazon EMR détecte un événement de AWS grande envergure dans une ou plusieurs zones de disponibilité, Amazon EMR tente automatiquement d'acheminer le trafic hors des zones de disponibilité concernées et essaie de lancer de nouveaux clusters que vous créez dans d'autres zones de disponibilité en fonction de vos sélections. Notez que la sélection de la zone de disponibilité du cluster s'effectue uniquement lors de la création du cluster.

Les nœuds de cluster existants ne sont pas automatiquement relancés dans une nouvelle zone de disponibilité en cas de panne de la zone de disponibilité.

Considérations

Tenez compte des éléments suivants lorsque vous utilisez des parcs d'instances avec Amazon EMR.

- Vous pouvez avoir un seul parc d'instances, par type de nœud (principal, de noyau, de tâche). Vous pouvez spécifier jusqu'à cinq types d'instances Amazon EC2 pour chaque flotte du AWS Management Console (ou un maximum de 30 types par flotte d'instances lorsque vous créez un cluster à l'aide de l'API AWS CLI Amazon EMR et d'un). Stratégie d'allocation pour les flottes d'instance
- Amazon EMR choisit un ou tous les types d'instance Amazon EC2 spécifiés pour allouer avec les options d'achat Spot et à la demande.
- Vous pouvez établir des capacités cibles pour les instances Spot et à la demande pour le parc principal et le parc de tâches. Utilisez un vCPU ou une unité générique attribuée à chaque instance Amazon EC2 prise en compte dans les cibles. Amazon EMR met en service des instances jusqu'à ce que chaque capacité cible soit atteinte. Pour le parc principal, la cible est toujours un.
- Vouz pouvez choisir un sous-réseau (zone de disponibilité) ou une plage. Si vous choisissez une fourchette, Amazon EMR fournit la capacité dans la zone de disponibilité la mieux adaptée.
- Lorsque vous spécifiez une capacité cible pour instances Spot :
  - Pour chaque type d'instance, spécifiez un prix Spot maximum. Amazon EMR met en service les instances Spot si le prix Spot est inférieur au prix Spot maximum. Vous payez le prix Spot, et non le prix Spot maximum.
  - 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 résilier le cluster ou passer à l'allocation de capacité à la demande à la place. Cela s'applique uniquement au provisionnement des clusters, et non à leur redimensionnement. Si le délai d'expiration prend fin pendant le processus de redimensionnement du cluster, les demandes Spot non provisionnées seront annulées sans être transférées vers la capacité à la demande.
- Pour chaque parc, vous pouvez définir l'une des stratégies d'allocation suivantes pour vos instances Spot : optimisation du rapport prix-capacité, optimisation de la capacité, prix le plus bas ou diversification sur tous les pools.
- Pour chaque parc, vous pouvez appliquer une stratégie d'allocation au prix le plus bas pour vos instances à la demande ; vous ne pouvez pas personnaliser la stratégie d'allocation pour les instances à la demande.

- Pour chaque parc avec `allocation strategy - lowest-price` à la demande, vous pouvez choisir d'appliquer des options de réserve de capacité.
- Vérifiez la taille de votre sous-réseau avant de lancer votre cluster. Lorsque vous approvisionnez un cluster avec un parc de tâches et qu'il n'y a pas suffisamment d'adresses IP disponibles dans le sous-réseau correspondant, le parc passe en état suspendu au lieu de résilier le cluster avec une erreur. Pour éviter ce problème, nous vous recommandons d'augmenter le nombre d'adresses IP dans vos sous-réseaux.

Options de parc d'instances

Suivez les instructions suivantes pour comprendre les options de parc d'instances.

Rubriques

- Définition des capacités cibles
- Options de lancement
- Options de sous-réseau multiples (zones de disponibilité)
- Configuration de nœud principal

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 principal 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 Amazon 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 le AWS CLI, vous attribuez manuellement des unités génériques à chaque type d'instance.

> **Important**
>
> Lorsque vous choisissez un type d'instance à l'aide du AWS Management Console, le nombre de vCPU indiqué pour chaque type d'instance est le nombre de vcores YARN

pour ce type d'instance, et non le nombre de vCPU EC2 pour ce type d'instance. Pour plus d'informations sur le nombre de vCPU pour chaque type d'instance, consultez Types d'instances Amazon EC2.

Vous pouvez spécifier jusqu'à cinq types d'instance Amazon EC2 pour chaque parc. Si vous utilisez Stratégie d'allocation pour les flottes d'instance et créez un cluster à l'aide de l'API AWS CLI ou de l'API Amazon EMR, vous pouvez spécifier jusqu'à 30 types d'instances EC2 par parc d'instances. Amazon EMR choisit une combinaison de ces types d'instances EC2 pour atteindre 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.

Options de lancement

Pour les instances Spot, vous pouvez spécifier un Prix Spot maximum pour chaque type d'instance dans un parc d'instances. Vous pouvez définir ce prix sous la forme d'un pourcentage du prix à la demande ou sous la forme d'un montant spécifique en dollars. Amazon EMR met en service les instances Spot si le prix Spot actuel dans une zone de disponibilité est inférieur à votre prix Spot maximum. Vous payez le prix Spot, et non le prix Spot maximum.

> **Note**
>
> Les instances Spot de durée définie (également appelées blocs d'instances Spot) ne sont plus disponibles pour les nouveaux clients depuis le 1er juillet 2021. Pour les clients qui ont déjà utilisé cette fonctionnalité, nous continuerons à prendre en charge les instances Spot de durée définie jusqu'au 31 décembre 2022.

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 l'ancienne version AWS Management Console ou à l'aide de l'API`RunJobFlow`. Notez que vous ne pouvez pas personnaliser la stratégie d'allocation dans la nouvelle console. L'utilisation de l'option de stratégie d'allocation nécessite des autorisations

de rôle de service supplémentaires. Si vous utilisez le rôle de service Amazon EMR par défaut et la politique gérée (`EMR_DefaultRole` et `AmazonEMRServicePolicy_v2`) pour le cluster, les autorisations pour l'option de stratégie d'allocation sont déjà incluses. Si vous n'utilisez pas le rôle de service Amazon EMR et la politique gérée par défaut, vous devez les ajouter pour utiliser cette option. veuillez consulter Rôle de service pour Amazon EMR (rôle EMR).

Pour plus d'informations sur les instances Spot, consultez la section Instances Spot dans le guide de l'utilisateur Amazon EC2. Pour plus d'informations sur les instances à la demande, consultez la section Instances à la demande dans le guide de l'utilisateur Amazon EC2.

Si vous choisissez de lancer des parcs d'instances à la demande avec la stratégie d'allocation des prix les plus bas, vous avez la possibilité d'utiliser les réservations de capacité. Les options de réserve de capacité peuvent être définies à l'aide de l'API Amazon EMR `RunJobFlow`. Les réservations de capacité nécessitent des autorisations de rôle de service supplémentaires que vous devez ajouter pour utiliser ces options. veuillez consulter Autorisations de la stratégie d'allocation. Notez que vous ne pouvez pas personnaliser les réservations de capacité dans la nouvelle console.

Options de sous-réseau multiples (zones de disponibilité)

Lorsque vous utilisez plusieurs parcs d'instances, vous pouvez spécifier plusieurs sous-réseaux Amazon EC2 au sein d'un VPC, chacun correspondant à une zone de disponibilité différente. Si vous utilisez EC2-Classic, vous spécifiez les zones de disponibilité de manière explicite. 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 le AWS CLI, réglez toujours l'un `TargetSpotCapacity` ou l'autre `TargetOnDemandCapacity` sur 1, selon le cas. Vous pouvez toujours choisir jusqu'à cinq types d'instances pour le parc d'instances principal (ou un maximum de 30 lorsque vous utilisez l'option de stratégie d'allocation pour les instances à la demande ou Spot). Cependant, contrairement aux parcs d'instances de noyau et de tâche, où Amazon EMR peut allouer

plusieurs instances de différents types, Amazon EMR sélectionne un seul type d'instance à allouer pour le parc d'instances principal.

Stratégie d'allocation pour les flottes d'instance

Avec les versions 5.12.1 et ultérieures d'Amazon EMR, vous pouvez utiliser l'option de stratégie d'allocation avec des instances à la demande et Spot pour chaque nœud de cluster. Lorsque vous créez un cluster à l'aide de l' AWS CLI, de l'API Amazon EMR ou de la console Amazon EMR avec une stratégie d'allocation, vous pouvez spécifier jusqu'à 30 types d'instances Amazon EC2 par parc. Avec la configuration par défaut du parc d'instances de cluster Amazon EMR, vous pouvez avoir jusqu'à 5 types d'instances par parc. Nous vous recommandons d'utiliser l'option de stratégie d'allocation pour accélérer le provisionnement des clusters, pour une allocation plus précise des instances Spot et pour réduire les interruptions des instances Spot.

Rubriques

- Stratégie d'allocation avec les instances à la demande
- Stratégie d'allocation avec les instances Spot
- Autorisations de la stratégie d'allocation
- Autorisations IAM requises pour une stratégie d'allocation

Stratégie d'allocation avec les instances à la demande

Lorsque vous utilisez une stratégie d'allocation, vos Instances à la demande utilisent la stratégie du prix le plus bas. Cela lance d'abord les instances les moins chères. Lorsque vous lancez des instances à la demande, vous pouvez utiliser des réserves de capacité ouvertes ou ciblées dans vos comptes. Vous pouvez utiliser des réserves de capacité ouvertes pour les nœuds primaires, les nœuds du noyau et les nœuds de tâches. Vous pouvez rencontrer une capacité insuffisante avec les instances à la demande dotées d'une stratégie d'allocation pour les parcs d'instances. Nous vous recommandons de spécifier un plus grand nombre de types d'instances afin de diversifier et de réduire les risques d'insuffisance de capacité. Pour plus d’informations, consultez Utilisation des réserves de capacité avec les parcs d'instances.

Stratégie d'allocation avec les instances Spot

Pour Instances Spot, vous pouvez choisir l'une des stratégies d'allocation suivantes :

`price-capacity-optimized` (recommandé)

La stratégie d'allocation optimisée en fonction du prix et de la capacité lance les instances Spot à partir des pools d'instances Spot qui ont la plus grande capacité disponible et le prix le plus bas par rapport au nombre d'instances en cours de lancement. Par conséquent, la stratégie d'optimisation prix-capacité a généralement plus de chances d'obtenir une capacité ponctuelle et permet de réduire les taux d'interruption.

`capacity-optimized`

La stratégie d'allocation optimisée en termes de capacité permet d'intégrer les instances Spot dans les pools les plus disponibles avec le moins de risques d'interruption à court terme. Il s'agit d'une bonne option pour les charges de travail dont le coût d'interruption peut être plus élevé en raison du redémarrage du travail. Il s'agit de la stratégie par défaut pour les versions 6.9.0 et antérieures d'Amazon EMR.

`diversified`

Grâce à sa stratégie d'allocation diversifiée, Amazon EC2 distribue des instances Spot dans tous les pools de capacité Spot.

`lowest-price`

La stratégie d'allocation au prix le plus bas lance les instances Spot à partir du pool le moins cher disposant de la capacité disponible. Si le groupe le moins coûteux ne dispose pas de capacité, les instances Spot proviennent du groupe le moins coûteux suivant qui a une capacité disponible. Si un groupe manque de capacité avant de répondre à votre demande, la flotte d'Amazon EC2 puise dans le groupe suivant dont le prix est le plus bas pour continuer à répondre à votre demande. Pour garantir que la capacité souhaitée est atteinte, vous pouvez recevoir des instances Spot de plusieurs groupes. Comme cette stratégie ne tient compte que du prix de l'instance et non de la disponibilité de la capacité, elle peut entraîner des taux d'interruption élevés.

Autorisations de la stratégie d'allocation

L'option de stratégie d'allocation nécessite plusieurs autorisations IAM qui sont automatiquement incluses dans le rôle de service Amazon EMR par défaut et dans la politique gérée par Amazon EMR (`EMR_DefaultRole` et `AmazonEMRServicePolicy_v2`). Si vous utilisez un rôle de service personnalisé ou une politique gérée pour votre cluster, vous devez ajouter ces autorisations avant de créer le cluster. Pour plus d'informations, consultez Autorisations de la stratégie d'allocation.

Les réservations de capacité à la demande (ODCR) facultatives sont disponibles lorsque vous utilisez l'option de stratégie d'allocation à la demande. Les options de réservation de capacité vous permettent de définir une préférence pour utiliser d'abord la capacité réservée pour les clusters Amazon EMR. Vous pouvez l'utiliser pour vous assurer que vos charges de travail critiques utilisent la capacité que vous avez déjà réservée à l'aide d'ODCR ouverts ou ciblés. Pour les charges de travail non critiques, les préférences de réservation de capacité vous permettent de spécifier si la capacité réservée doit être consommée.

Les réserves de capacité ne peuvent être utilisées que par des instances qui correspondent à leurs attributs (type d'instance, plateforme et zone de disponibilité). Par défaut, les réservations de capacité ouverte sont automatiquement utilisées par Amazon EMR lors du provisionnement d'instances à la demande qui correspondent aux attributs de l'instance. Si aucune instance en cours d'exécution ne correspond aux attributs des réserves de capacité, celles-ci restent inutilisées jusqu'à ce que vous lanciez une instance correspondant à leurs attributs. Si vous ne voulez utiliser aucune réserve de capacité lors du lancement de votre cluster, vous devez définir la préférence de réserve de capacité sur none dans les options de lancement.

Cependant, vous pouvez également cibler une Réservation de capacité pour des charges de travail spécifiques. Cela vous permet de contrôler explicitement les instances autorisées à s'exécuter dans cette capacité réservée. Pour plus d'informations sur les réserves de capacité à la demande, consultez Utilisation des réserves de capacité avec les parcs d'instances.

Autorisations IAM requises pour une stratégie d'allocation

Votre Rôle de service pour Amazon EMR (rôle EMR) a besoin d'autorisations supplémentaires pour créer un cluster qui utilise l'option de stratégie d'allocation pour les parcs d'instances à la demande ou Spot.

Nous incluons automatiquement ces autorisations dans le rôle de service Amazon EMR par défaut `EMR_DefaultRole` et dans la politique gérée par Amazon EMR `AmazonEMRServicePolicy_v2`.

Si vous utilisez un rôle de service personnalisé ou une politique gérée pour votre cluster, vous devez ajouter les autorisations suivantes :

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

```
   "Action": [
    "ec2:DeleteLaunchTemplate",
    "ec2:CreateLaunchTemplate",
    "ec2:DescribeLaunchTemplates",
    "ec2:CreateLaunchTemplateVersion",
    "ec2:CreateFleet"
   ],
   "Resource": "*"
  }
}
```

Les autorisations de rôle de service suivantes sont requises pour créer un cluster qui utilise des réservations de capacité ouvertes ou ciblées. Vous devez inclure ces autorisations en plus des autorisations requises pour utiliser l'option de stratégie d'allocation.

Example Document de politique pour les réservations de capacité des rôles de service

Pour utiliser les réservations de capacité ouverte, vous devez inclure les autorisations supplémentaires suivantes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

Example

Pour utiliser les réservations de capacité ciblées, vous devez inclure les autorisations supplémentaires suivantes.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions",
                "resource-groups:ListGroupResources"
            ],
            "Resource": "*"
        }
    ]
}
```

Configurer des parcs d'instances pour votre cluster

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour créer un cluster avec des parcs d'instances à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le panneau de navigation de gauche, sélectionnez Clusters, puis Créer un cluster.
3. Sous Configuration du cluster, choisissez Parcs d'instances.
4. Pour chaque Groupe de nœuds, sélectionnez Ajouter un type d'instance et choisissez jusqu'à 5 types d'instances pour les parcs d'instances principales et de noyau et jusqu'à quinze types d'instances pour les parcs d'instances de tâches. Amazon EMR peut allouer n'importe quelle combinaison de ces types d'instance lorsqu'il lance le cluster.
5. Sous chaque type de groupe de nœuds, choisissez le menu déroulant Actions en regard de chaque instance pour modifier ces paramètres :

Ajoutez des volumes EBS.

Spécifiez les volumes EBS à attacher au type d'instance après le provisionnement par Amazon EMR.

Modifier la capacité pondérée

Pour le groupe de nœuds principaux, remplacez cette valeur par un nombre d'unités adapté à vos applications. Le nombre de vCores YARN pour chaque type d'instance de parc est utilisé comme unité de capacité pondérée par défaut. Vous ne pouvez pas modifier la capacité pondérée du nœud primaire.

Modification du prix Spot maximum

Spécifiez un prix Spot maximum pour chaque type d'instances dans un parc instances. Vous pouvez définir ce prix sous la forme d'un pourcentage du prix à la demande ou sous la forme d'un montant spécifique en dollars. Si le prix Spot actuel dans une zone de disponibilité est inférieur à votre prix Spot maximum, Amazon EMR met en service des instances Spot. Vous payez le prix Spot, et non le prix Spot maximum.

6. Vous pouvez éventuellement ajouter des groupes de sécurité pour vos nœuds, développez Groupes de sécurité EC2 (pare-feu) dans la section Mise en réseau et sélectionnez votre groupe de sécurité pour chaque type de nœud.
7. Cochez éventuellement la case à côté de Appliquer la stratégie d'allocation si vous souhaitez utiliser l'option de stratégie d'allocation, puis sélectionnez la stratégie d'allocation que vous souhaitez spécifier pour les instances Spot. Vous ne devez pas sélectionner cette option si votre rôle de service Amazon EMR ne dispose pas des autorisations requises. Pour plus d'informations, consultez Stratégie d'allocation pour les flottes d'instance.
8. Choisissez toutes les autres options qui s'appliquent à votre cluster.
9. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour créer un cluster avec des parcs d'instances avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster.

3. En haut de la fenêtre de console, choisissez Accéder aux options avancées, entrez les options de Configuration logicielle, puis cliquez sur Suivant.
4. Sous Composition du cluster, sélectionnez Parcs d'instances. Lorsque vous sélectionnez l'option parcs d'instances, les options permettant de spécifier la Capacité cible des instances à la demande et Spot devraient apparaître dans le tableau des Nœuds et instances du cluster.
5. 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 Amazon 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 plus d'informations, consultez Amazon EC2 et Amazon VPC dans le guide de l'utilisateur Amazon EC2.

6. Sous Stratégie d'allocation, cochez la case pour appliquer les stratégies d'allocation si vous souhaitez utiliser l'option de stratégie d'allocation. Pour plus d'informations, consultez Stratégie d'allocation pour les flottes d'instance.
7. Pour chaque Type de nœud, si vous voulez 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 parc d'instances de Tâche, cliquez sur l'icône X sur le côté droit de la ligne de tâches.
8. Choisissez Ajouter/supprimer des types d'instances au parc et choisissez jusqu'à cinq types d'instances dans la liste pour les parcs d'instances principales et principales ; ajoutez jusqu'à quinze types d'instances pour les parcs d'instances de tâches. Amazon EMR peut choisir de provisionner n'importe quelle combinaison de ces types d'instances lors du lancement du cluster.
9. Pour chaque type d'instance principale et de tâche, choisissez la manière dont vous souhaitez définir la capacité pondérée (chaque instance compte pour X unités) pour cette instance. Le nombre de YARN VCores pour chaque type de parc d'instance est utilisé comme

unité de capacité pondérée par défaut, mais vous pouvez modifier la valeur en n'importe quelle unité adaptée à vos applications.

10. Sous Capacité cible, définissez le nombre total d'instances à la demande et Spot que vous souhaitez par parc. L'EMR garantit que les instances du parc fournissent les unités demandées pour la capacité cible à la demande et Spot. Si aucune unité à la demande ou Spot n'est spécifiée pour une parc, aucune capacité n'est allouée pour cette parc.
11. Si un parc d'instances est configuré 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.
12. Pour que les volumes EBS soient attachés au type d'instance lors de sa mise en service, choisissez le crayon à côté de Stockage sur EBS et indiquez les options de configuration EBS.
13. Si vous avez établi un décompte instantané pour les Unités Spot, définissez les Options Spot avancées conformément aux directives suivantes :

    - 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 passer à 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 des instances à la demande jusqu'à ce que la capacité cible des instances Spot soit atteinte.
14. Choisissez Suivant, modifiez les autres paramètres du cluster, puis choisissez Suivant.
15. Si vous avez choisi d'appliquer la nouvelle option de stratégie d'allocation, dans les paramètres des Options de sécurité, sélectionnez un Rôle EMR et un Profil d'instance EC2 contenant les autorisations requises pour l'option de stratégie d'allocation. Dans le cas contraire, la création du cluster échouera.
16. Choisissez Create Cluster (Créer un cluster).

AWS CLI

Pour créer et lancer un cluster avec des flottes d'instances dotées du AWS CLI, suivez les instructions suivantes :

- 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 sur les parcs d'instances dans un cluster, utilisez la commande `list-instance-fleets`.
- Pour ajouter plusieurs AMI Amazon Linux personnalisées à un cluster que vous créez, utilisez l'option `CustomAmiId` associée à chaque spécification `InstanceType`. Vous pouvez configurer des nœuds de parc d'instances avec plusieurs types d'instances et plusieurs AMI personnalisées pour répondre à vos besoins. veuillez consulter Exemples : Création d'un cluster avec la configuration de parcs d'instances.
- 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`.
- Plusieurs AMI personnalisées peuvent être ajoutées au parc d'instances de tâches à l'aide de l'CustomAmiId argument associé à la add-instance-fleet commande. veuillez consulter Exemples : Création d'un cluster avec la configuration de parcs d'instances.
- 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 politique dans la section suivante.
- Pour utiliser les options de réservation de capacité lors de la création d'un parc d'instances avec une stratégie d'allocation à la demande, mettez à jour le rôle de service pour inclure l'exemple de document de politique dans la section suivante.
- Les parcs d'instances sont automatiquement incluses dans le rôle de service EMR par défaut et dans la politique gérée par Amazon EMR (`EMR_DefaultRole` et `AmazonEMRServicePolicy_v2`). Si vous utilisez un rôle de service personnalisé ou une politique gérée par le client pour votre cluster, vous devez ajouter les nouvelles autorisations pour la stratégie d'allocation dans la section suivante.

Exemples : 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 Amazon 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 : principal à la demande, base à la demande avec un 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.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}
```

Example Exemple : Principal Spot, Spot de noyau avec un seul type d'instance, 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=0.5}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Example Exemple : principal à la demande, base 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=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=2}']
```

Example Exemple : Principal à la demande, Spot de noyau avec plusieurs types d'instances pondérées, délai d'attente pour Spot, gamme 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,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
```

Example Exemple : Principal à la demande, de noyau mixte et tâche avec plusieurs types d'instances pondérées, délai d'attente pour les instances Spot de noyau, 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.xlarg
 \
    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='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
 \
    InstanceFleetType=TASK,TargetOnDemandCapacity=3,TargetSpotCapacity=3,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}']
```

Example Exemple : Principal Spot, pas de noyau ou tâche, configuration Amazon EBS, VPC par défaut.

```
aws emr create-cluster --release-label Amazon 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_CLU
\
```

```
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 Exemple : plusieurs AMI personnalisées, plusieurs types d'instances, principale à la demande, cœur à la demande

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole
 \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}']
```

Example Exemple : ajouter un nœud de tâche à un cluster en cours d'exécution avec plusieurs types d'instances et plusieurs AMI personnalisées

```
aws emr add-instance-fleet --cluster-id j-123456 --release-label Amazon EMR 5.3.1 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleet \
    InstanceFleetType=Task,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',\
'{InstanceType=m6g.xlarge,CustomAmiId=ami-234567}']
```

Example Exemple : Utiliser 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 fait référence à 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
```

Le fichier *my-fleet-config.json* spécifie les parcs d'instances principales, de noyau et de tâche, comme indiqué dans l'exemple suivant. Le parc d'instances principal utilise un prix spot maximum (`BidPrice`) en pourcentage de la demande, tandis que les flottes de tâches et d'instances principales utilisent un prix spot maximum (BidPriceAsPercentageofOnDemandPrice) sous forme de chaîne 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",
            "CapacityReservationOptions":
            {
                "UsageStrategy": "use-capacity-reservations-first",
                "CapacityReservationResourceGroupArn": "String"
            }
        },
            "SpotSpecification": {
                "AllocationStrategy": "capacity-optimized",
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
```

```
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPriceAsPercentageOfOnDemandPrice": 100
            }
        ]
    },
    {
        "Name": "Taskfleet",
        "InstanceFleetType": "TASK",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions":
            {
                "CapacityReservationPreference": "none"
            }
        },
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    }
]
```

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 <cluster-id> \
  --instance-fleet \
```

```
    InstanceFleetId='<instance-fleet-id>',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 <cluster-id>
  --instance-fleet \
    InstanceFleetType=TASK,TargetSpotCapacity=1,\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=20,TimeoutAction=TERMINATE_CLU
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

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 connaître la syntaxe complète des réponses, consultez ListInstanceFleets dans le manuel Amazon EMR API Reference.

```
list-instance-fleets --cluster-id <cluster-id>
```

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

Utilisation des réserves de capacité avec les parcs d'instances

Pour lancer des parcs d'instances à la demande avec des options de réservation de capacité, associez les autorisations de rôle de service supplémentaires requises pour utiliser les options de réservation de capacité. Étant donné que les options de réservation de capacité doivent être utilisées conjointement avec la stratégie d'allocation à la demande, vous devez également inclure les autorisations requises pour la stratégie d'allocation dans votre rôle de service et votre politique gérée. Pour plus d'informations, consultez Autorisations de la stratégie d'allocation.

Amazon EMR prend en charge les réservations de capacité ouvertes et ciblées. Les rubriques suivantes présentent les configurations de parcs d'instances que vous pouvez utiliser avec l'action `RunJobFlow` ou la commande `create-cluster` pour lancer des parcs d'instances à l'aide de réserves de capacité à la demande.

Utilisez les réservations de capacité ouverte au mieux

Si les instances à la demande du cluster correspondent aux attributs des réserves de capacité ouvertes (type d'instance, plateforme, location et zone de disponibilité) disponibles dans votre compte, les réserves de capacité sont appliquées automatiquement. Cependant, il n'est pas garanti que vos réservations de capacité seront utilisées. Pour le provisionnement du cluster, Amazon EMR évalue tous les pools d'instances spécifiés dans la demande de lancement et utilise celui dont le prix le plus bas possède une capacité suffisante pour lancer tous les nœuds principaux demandés. Les réserves de capacité ouvertes disponibles qui correspondent au groupe d'instances sont appliquées automatiquement. Si les réserves de capacité ouvertes disponibles ne correspondent pas au groupe d'instances, elles restent inutilisées.

Une fois les nœuds principaux alloués, la zone de disponibilité est sélectionnée et fixée. Amazon EMR approvisionne les nœuds de tâches dans des pools d'instances, en commençant par les nœuds les moins chers, dans la zone de disponibilité sélectionnée jusqu'à ce que tous les nœuds de tâches soient provisionnés. Les réservations de capacité ouverte disponibles qui correspondent aux pools d'instances sont appliquées automatiquement.

Vous trouverez ci-dessous des exemples d'utilisation de la logique d'allocation de capacité Amazon EMR pour utiliser au mieux les réservations de capacité ouverte.

Exemple 1 : le groupe d'instances au prix le plus bas indiqué dans la demande de lancement dispose de réserves de capacité ouvertes

Dans ce cas, Amazon EMR lance la capacité dans le groupe d'instances le moins cher avec des instances à la demande. Vos réservations de capacité ouverte disponibles dans ce groupe d'instances sont utilisées automatiquement.

Stratégie à la demande	lowest-price (prix le plus bas)		
Capacité demandée	100		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge
Réserves de capacité ouvertes disponibles	150	100	100
Prix à la demande	$	$$	$$$
Instances allouées	100	-	-
Réserves de capacité ouvertes utilisées	100	-	-
Réserves de capacité ouvertes disponibles	50	100	100

Une fois le parc d'instances lancé, vous pouvez exécuter `describe-capacity-reservations` pour voir combien de réserves de capacité inutilisées restent.

Exemple 2 : le groupe d'instances au prix le plus bas indiqué dans la demande de lancement ne dispose pas de réserves de capacité ouvertes disponibles

Dans ce cas, Amazon EMR lance la capacité dans le groupe d'instances le moins cher avec des instances à la demande. Cependant, vos réservations de capacité ouverte restent inutilisées.

Stratégie à la demande	lowest-price (prix le plus bas)		
Capacité demandée	100		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge

Réserves de capacité ouvertes disponibles	-	-	100
Prix à la demande	$	$$	$$$
Instances allouées	100	-	-
Réserves de capacité ouvertes utilisées	-	-	-
Réserves de capacité ouvertes disponibles	-	-	100

Configurer les parcs d'instances pour utiliser au mieux les réserves de capacité ouverte

Lorsque vous utilisez l'action `RunJobFlow` pour créer un cluster basé sur un parc d'instances, définissez la stratégie d'allocation à la demande sur `lowest-price` et `CapacityReservationPreference` pour les options de réserve de capacité sur `open`. Sinon, si vous laissez ce champ vide, Amazon EMR définit par défaut la préférence de réserve de capacité de l'instance à la demande sur `open`.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "open"
         }
        }
    }
```

Vous pouvez également utiliser la CLI Amazon EMR pour créer un cluster basé sur un parc d'instances à l'aide de réservations de capacité ouverte.

```
aws emr create-cluster \
 --name 'open-ODCR-cluster' \
 --release-label emr-5.30.0 \
 --service-role EMR_DefaultRole \
 --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
```

```
 --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
   LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=open}}'}
```

Où,

- `open-ODCR-cluster` est remplacé par le nom du cluster utilisant des réserves de capacité ouvertes.
- `subnet-22XXXX01` est remplacé par l'ID du sous-réseau.

Utilisez d'abord les réservations de capacité ouverte

Vous pouvez choisir d'annuler la stratégie d'allocation du prix le plus bas et de prioriser l'utilisation des réservations de capacité ouverte disponibles en premier lors de la mise en service d'un cluster Amazon EMR. Dans ce cas, Amazon EMR évalue tous les pools d'instances dont les réservations de capacité sont spécifiées dans la demande de lancement et utilise celui dont le prix le plus bas possède une capacité suffisante pour lancer tous les nœuds principaux demandés. Si aucun des pools d'instances avec des réservations de capacité ne dispose d'une capacité suffisante pour les nœuds principaux demandés, Amazon EMR revient au scénario de « meilleur effort » décrit dans la rubrique précédente. C'est-à-dire qu'Amazon EMR réévalue tous les pools d'instances spécifiés dans la demande de lancement et utilise celui dont le prix le plus bas possède une capacité suffisante pour lancer tous les nœuds principaux demandés. Les réserves de capacité ouvertes disponibles qui correspondent au groupe d'instances sont appliquées automatiquement. Si les réserves de capacité ouvertes disponibles ne correspondent pas au groupe d'instances, elles restent inutilisées.

Une fois les nœuds principaux alloués, la zone de disponibilité est sélectionnée et fixée. Amazon EMR approvisionne les nœuds de tâches dans des pools d'instances avec des réservations de capacité, en commençant par les nœuds les moins chers, dans la zone de disponibilité sélectionnée jusqu'à ce que tous les nœuds de tâches soient approvisionnés. Amazon EMR utilise d'abord les réservations de capacité ouverte disponibles sur chaque groupe d'instances de la zone de disponibilité sélectionnée, et uniquement si nécessaire, utilise la stratégie du prix le plus bas pour approvisionner les nœuds de tâches restants.

Vous trouverez ci-dessous des exemples d'utilisation de la logique d'allocation de capacité Amazon EMR pour utiliser d'abord les réservations de capacité ouverte.

Exemple 1 : le groupe d'instances avec des réserves de capacité ouvertes disponibles dans la demande de lancement dispose d'une capacité suffisante pour les nœuds principaux

Dans ce cas, Amazon EMR lance la capacité dans le groupe d'instances avec les réservations de capacité ouverte disponibles, quel que soit le prix du groupe d'instances. Par conséquent, vos réservations de capacité ouverte sont utilisées dans la mesure du possible, jusqu'à ce que tous les nœuds principaux soient approvisionnés.

Stratégie à la demande	lowest-price (prix le plus bas)		
Capacité demandée	100		
Stratégie d'utilisation	utilisation-capacité-réservations-first		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge
Réserves de capacité ouvertes disponibles	-	-	150
Prix à la demande	$	$$	$$$
Instances allouées	-	-	100
Réserves de capacité ouvertes utilisées	-	-	100
Réserves de capacité ouvertes disponibles	-	-	50

Exemple 2 : le groupe d'instances avec des réserves de capacité ouvertes disponibles dans la demande de lancement ne dispose pas d'une capacité suffisante pour les nœuds principaux

Dans ce cas, Amazon EMR se contente de lancer des nœuds principaux en utilisant la stratégie du prix le plus bas tout en s'efforçant d'utiliser les réservations de capacité.

Stratégie à la demande	lowest-price (prix le plus bas)		
Capacité demandée	100		
Stratégie d'utilisation	utilisation-capacité-réservations-first		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge
Réserves de capacité ouvertes disponibles	10	50	50
Prix à la demande	$	$$	$$$
Instances allouées	100	-	-
Réserves de capacité ouvertes utilisées	10	-	-
Réservations de capacité ouverte disponibles	-	50	50

Une fois le parc d'instances lancé, vous pouvez exécuter `describe-capacity-reservations` pour voir combien de réserves de capacité inutilisées restent.

Configurer les parcs d'instances pour utiliser d'abord les réserves de capacité ouvertes

Lorsque vous utilisez l'action `RunJobFlow` pour créer un cluster basé sur un parc d'instances, définissez la stratégie d'allocation à la demande sur `lowest-price` et `UsageStrategy` pour `CapacityReservationOptions` sur `use-capacity-reservations-first`.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first"
         }
       }
```

```
    }
```

Vous pouvez également utiliser la CLI Amazon EMR pour créer un cluster basé sur un parc d'instances en utilisant d'abord les réservations de capacité.

```
aws emr create-cluster \
  --name 'use-CR-first-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-first}}'}
```

Où,

- `use-CR-first-cluster` est remplacé par le nom du cluster utilisant des réserves de capacité ouvertes.
- `subnet-22XXXX01` est remplacé par l'ID du sous-réseau.

Utilisez d'abord des réservations de capacité ciblées

Lorsque vous mettez en service un cluster Amazon EMR, vous pouvez choisir d'annuler la stratégie d'allocation du prix le plus bas et de prioriser l'utilisation des réservations de capacité ciblées disponibles en premier. Dans ce cas, Amazon EMR évalue tous les pools d'instances avec des réservations de capacité ciblées spécifiées dans la demande de lancement et choisit celle dont le prix le plus bas possède une capacité suffisante pour lancer tous les nœuds principaux demandés. Si aucun des pools d'instances avec des réservations de capacité ciblées ne dispose d'une capacité suffisante pour les nœuds principaux, Amazon EMR revient au scénario de meilleur effort décrit précédemment. C'est-à-dire qu'Amazon EMR réévalue tous les pools d'instances spécifiés dans la demande de lancement et sélectionne celui dont le prix le plus bas possède une capacité suffisante pour lancer tous les nœuds principaux demandés. Les réservations de capacité ouverte disponibles

qui correspondent au groupe d'instances sont appliquées automatiquement. Cependant, les réservations de capacité ciblées restent inutilisées.

Une fois les nœuds principaux alloués, la zone de disponibilité est sélectionnée et fixée. Amazon EMR approvisionne les nœuds de tâches dans des pools d'instances avec des réservations de capacité ciblées, en commençant par les nœuds les moins chers, dans la zone de disponibilité sélectionnée jusqu'à ce que tous les nœuds de tâches soient approvisionnés. Amazon EMR essaie d'abord d'utiliser les réservations de capacité ciblées disponibles sur chaque groupe d'instances de la zone de disponibilité sélectionnée. Ensuite, uniquement si nécessaire, Amazon EMR utilise la stratégie du prix le plus bas pour approvisionner les nœuds de tâches restants.

Vous trouverez ci-dessous des exemples d'utilisation de la logique d'allocation de capacité Amazon EMR pour utiliser d'abord des réservations de capacité ciblées.

Exemple 1 : le groupe d'instances pour lequel des réserves de capacité ciblées sont disponibles dans la demande de lancement dispose d'une capacité suffisante pour les nœuds principaux

Dans ce cas, Amazon EMR lance de la capacité dans le groupe d'instances avec des réservations de capacité ciblées disponibles, quel que soit le prix du groupe d'instances. Par conséquent, vos réservations de capacité ciblées sont utilisées dans la mesure du possible jusqu'à ce que tous les nœuds principaux soient approvisionnés.

Stratégie à la demande	lowest-price (prix le plus bas)		
Stratégie d'utilisation	utilisation-capacité-réservations-first		
Capacité demandée	100		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge
Réserves de capacité ciblées disponibles	-	-	150
Prix à la demande	$	$$	$$$
Instances allouées	-	-	100

Réservation de capacité ciblée utilisée	-	-	100
Réserves de capacité ciblées disponibles	-	-	50

Example Exemple 2 : le groupe d'instances pour lequel des réservations de capacité ciblées sont disponibles dans la demande de lancement ne dispose pas d'une capacité suffisante pour les nœuds principaux

Stratégie à la demande	lowest-price (prix le plus bas)		
Capacité demandée	100		
Stratégie d'utilisation	utilisation-capacité-réservations-first		
Type d'instance	c5.xlarge	m5.xlarge	r5.xlarge
Réserves de capacité ciblées disponibles	10	50	50
Prix à la demande	$	$$	$$$
Instances allouées	100	-	-
Réserves de capacité ciblées utilisées	10	-	-
Réserves de capacité ciblées disponibles	-	50	50

Une fois le parc d'instances lancé, vous pouvez exécuter `describe-capacity-reservations` pour voir combien de réserves de capacité inutilisées restent.

Configurer les parcs d'instances pour utiliser d'abord les réserves de capacité ciblées

Lorsque vous utilisez l'action `RunJobFlow` pour créer un cluster basé sur un parc d'instances, définissez la stratégie d'allocation à la demande sur `lowest-price`, `UsageStrategy` pour `CapacityReservationOptions` sur `use-capacity-reservations-first`, et `CapacityReservationResourceGroupArn` pour`CapacityReservationOptions` sur `<your resource group ARN>`. Pour plus d'informations, consultez la section Travailler avec les réservations de capacité dans le guide de l'utilisateur Amazon EC2.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first",
            "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup"
         }
       }
    }
```

Où `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` est remplacé par l'ARN de votre groupe de ressources.

Vous pouvez également utiliser l'interface de ligne de commande Amazon EMR pour créer un cluster basé sur un parc d'instances à l'aide de réservations de capacité ciblées.

```
aws emr create-cluster \
  --name 'targeted-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \
    InstanceFleetType=CORE,TargetOnDemandCapacity=100,\
InstanceTypeConfigs=['{InstanceType=c5.xlarge},{InstanceType=m5.xlarge},
{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-
first,CapacityReservationResourceGroupArn=arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup}}'}
```

Où,

- `targeted-CR-cluster` est remplacé par le nom de votre cluster à l'aide de réserves de capacité ciblées.
- `subnet-22XXXX01` est remplacé par l'ID du sous-réseau.
- `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` est remplacé par l'ARN de votre groupe de ressources.

Évitez d'utiliser les réservations de capacité libre disponibles

Example

Si vous souhaitez éviter d'utiliser de manière inattendue l'une de vos réserves de capacité ouverte lors du lancement d'un cluster Amazon EMR, définissez la stratégie d'allocation à la demande sur `lowest-price` et `CapacityReservationPreference` pour `CapacityReservationOptions` sur `none`. Dans le cas contraire, Amazon EMR définit par défaut la préférence de réserve de capacité de l'instance à la demande sur `open` et essaie d'utiliser au mieux les réserves de capacité ouverte disponibles.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "none"
         }
       }
    }
```

Vous pouvez également utiliser la CLI Amazon EMR pour créer un cluster basé sur un parc d'instances sans utiliser de réservations de capacité ouverte.

```
aws emr create-cluster \
  --name 'none-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
```

```
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=none}}'}
```

Où,

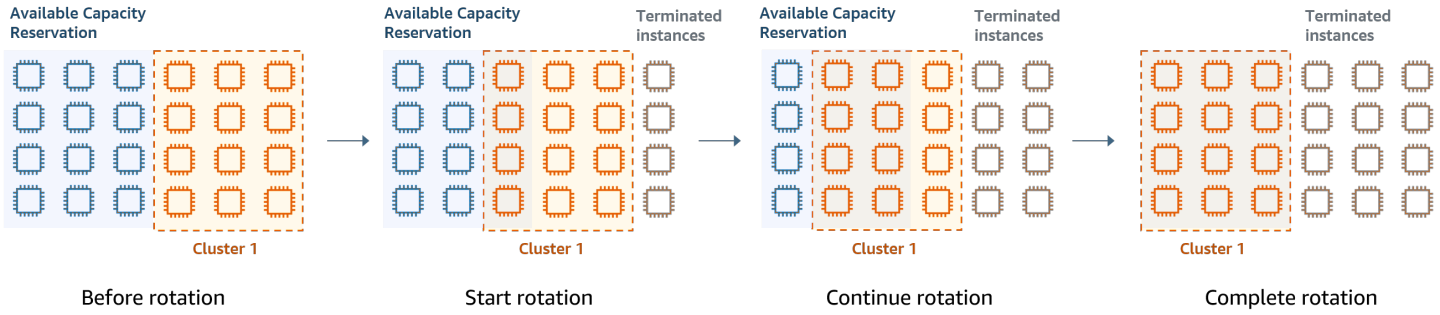
- `none-CR-cluster` est remplacé par le nom de votre cluster qui n'utilise aucune réserve de capacité ouverte.
- `subnet-22XXXX01` est remplacé par l'ID du sous-réseau.

Scénarios d'utilisation des réservations de capacité

Vous pouvez bénéficier de l'utilisation des réservations de capacité dans les scénarios suivants.

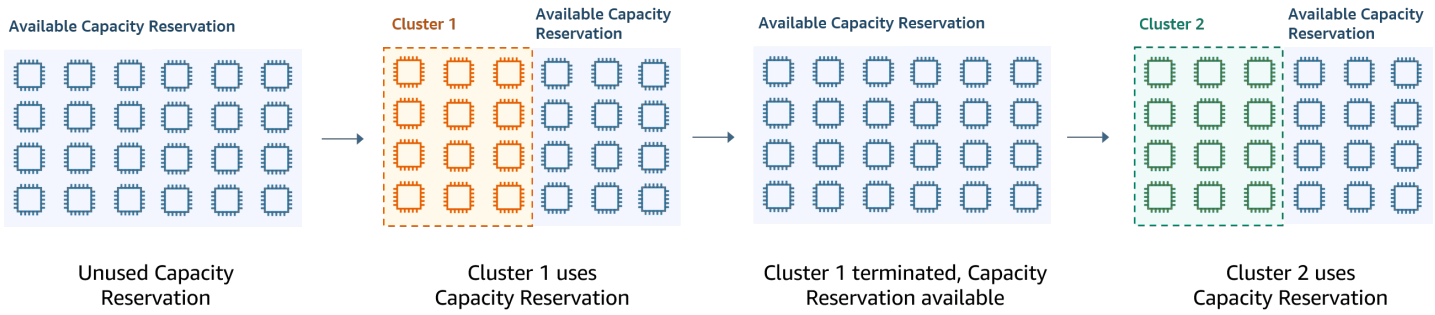
Scénario 1 : rotation d'un cluster de longue durée à l'aide de réservations de capacité

Lors de la rotation d'un cluster de longue durée, vous pouvez avoir des exigences strictes concernant les types d'instances et les zones de disponibilité pour les nouvelles instances que vous mettez en service. Avec les réservations de capacité, vous pouvez utiliser l'assurance de capacité pour effectuer la rotation du cluster sans interruption.



Scénario 2 : provisionner des clusters de courte durée successifs à l'aide de réservations de capacité

Vous pouvez également utiliser les réservations de capacité pour provisionner un groupe de clusters successifs de courte durée pour des charges de travail individuelles, de sorte que lorsque vous mettez fin à un cluster, le cluster suivant puisse utiliser les réservations de capacité. Vous pouvez utiliser des réservations de capacité ciblées pour vous assurer que seuls les clusters prévus utilisent les réservations de capacité.



Configuration de groupes d'instances uniformes

Avec la configuration des groupes d'instances, chaque 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.

Si les instances à la demande du cluster correspondent aux attributs des réserves de capacité ouvertes (type d'instance, plateforme, location et zone de disponibilité) disponibles dans votre compte, les réserves de capacité sont appliquées automatiquement. Vous pouvez utiliser des réserves de capacité ouvertes pour les nœuds primaires, les nœuds du noyau et les nœuds de tâches. Toutefois, vous ne pouvez pas utiliser de réservations de capacité ciblées ni empêcher les instances de se lancer dans des réservations de capacité ouverte avec des attributs correspondants lorsque vous provisionnez des clusters à l'aide de groupes d'instances. Si vous souhaitez utiliser des réservations de capacité ciblées ou empêcher les instances de se lancer dans des réservations de capacité ouverte, utilisez plutôt des parcs d'instances. Pour plus d’informations, consultez Utilisation des réserves de capacité avec les parcs d'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 Utiliser la mise à l'échelle des clusters.

Lors du lancement des instances, la préférence de réservation de capacité de l'instance à la demande est par défaut `open`, ce qui lui permet de s'exécuter dans toute réserve de capacité ouverte dont les attributs correspondent (type d'instance, plateforme, zone de disponibilité). Pour plus d'informations sur les réserves de capacité à la demande, consultez Utilisation des réserves de capacité avec les parcs d'instances.

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 Gestion des clusters.

Utilisation de la console pour configurer des groupes d'instances uniformes

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour créer un cluster avec des groupes d'instances avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le panneau de navigation de gauche, sélectionnez Clusters, puis Créer un cluster.
3. Sous Configuration du cluster, sélectionnez Groupes d'instances.
4. Sous Groupes de nœuds, il existe une section pour chaque type de groupe de nœuds. Pour le groupe de nœuds primaires, cochez la case Utiliser plusieurs nœuds primaires si vous souhaitez avoir 3 nœuds primaires. Cochez la case Utiliser l'option d'achat Spot si vous souhaitez utiliser l'option d'achat Spot.
5. Pour les groupes de nœuds principaux et principaux, sélectionnez Ajouter un type d'instance et choisissez jusqu'à 5 types d'instance. Pour le groupe de tâches, sélectionnez Ajouter un type d'instance et choisissez jusqu'à quinze types d'instance. Amazon EMR peut allouer n'importe quelle combinaison de ces types d'instance lorsqu'il lance le cluster.
6. Sous chaque type de groupe de nœuds, choisissez le menu déroulant Actions en regard de chaque instance pour modifier ces paramètres :

   Ajoutez des volumes EBS.

   Spécifiez les volumes EBS à attacher au type d'instance après le provisionnement par Amazon EMR.

Modification du prix Spot maximum

Spécifiez un prix Spot maximum pour chaque type d'instances dans un parc instances. Vous pouvez définir ce prix sous la forme d'un pourcentage du prix à la demande ou sous la forme d'un montant spécifique en dollars. Si le prix Spot actuel dans une zone de disponibilité est inférieur à votre prix Spot maximum, Amazon EMR met en service des instances Spot. Vous payez le prix Spot, et non le prix Spot maximum.

7. Vous pouvez éventuellement développez Configuration du nœud pour entrer une configuration JSON ou pour charger JSON depuis Amazon S3.
8. Choisissez toutes les autres options qui s'appliquent à votre cluster.
9. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

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 créer un cluster avec des groupes d'instances uniformes avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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 Configuration de la mise en réseau.

> **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 plus d'informations, consultez Amazon EC2 et Amazon VPC dans le guide de l'utilisateur Amazon EC2.

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, choisissez l'icône en forme de crayon et saisissez un nom convivial. Si vous voulez supprimer 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 Type d'instance, choisissez l'icône en forme de crayon, puis le type d'instance que vous voulez utiliser pour ce type de nœud.

     Important

     Lorsque vous choisissez un type d'instance à l'aide du AWS Management Console, le nombre de vCPU indiqué pour chaque type d'instance est le nombre de vcores YARN pour ce type d'instance, et non le nombre de vCPU EC2 pour ce type d'instance. Pour plus d'informations sur le nombre de vCPU pour chaque type d'instance, consultez Types d'instances Amazon EC2.

   - Sous Type d'instance, choisissez 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**
>
> Faites une pause sur l'info-bulle Spot pour voir le prix Spot actuel pour les zones de disponibilité dans la région actuelle. 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 Utilisation de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.

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.

Utilisez le 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 utilise l'option Instance 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 la Référence des commandes AWS CLI .

Vous pouvez utiliser le AWS CLI pour créer des groupes d'instances uniformes dans un cluster utilisant 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* par le nom de votre paire de clés Amazon EC2.

> **Note**
>
> Les caractères de continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "MySpotCluster" \
  --release-label emr-7.1.0 \
  --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
```

À l'aide de la CLI, vous pouvez créer des clusters de groupes d'instances uniformes qui spécifient une AMI personnalisée unique pour chaque type d'instance du groupe d'instances. Cela vous permet d'utiliser différentes architectures d'instance dans le même groupe d'instances. Chaque type d'instance doit utiliser une AMI personnalisée avec une architecture correspondante. Par exemple, vous devez configurer un type d'instance m5.xlarge avec une AMI personnalisée d'architecture x86_64, et un type d'instance m6g.xlarge avec une AMI personnalisée d'architecture `AWS  AARCH64` (ARM) correspondante.

L'exemple suivant montre un cluster de groupes d'instances uniforme créé avec deux types d'instances, chacun doté de sa propre AMI personnalisée. Notez que les AMI personnalisées sont spécifiées uniquement au niveau du type d'instance, et non au niveau du cluster. Cela permet d'éviter les conflits entre les AMI du type d'instance et une AMI au niveau du cluster, ce qui entraînerait l'échec du lancement du cluster.

```
aws emr create-cluster
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
```

```
  --instance-groups \

 InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
 \

 InstanceGroupType=CORE,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-234567
```

Vous pouvez ajouter plusieurs AMI personnalisées à un groupe d'instances que vous ajoutez à un cluster en cours d'exécution. L'argument `CustomAmiId` peut être utilisé avec la commande `add-instance-groups` comme le montre l'exemple suivant.

```
aws emr add-instance-groups --cluster-id j-123456 \
  --instance-groups \

 InstanceGroupType=Task,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
```

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");
```

Bonnes pratiques pour la flexibilité des instances et des zones de disponibilité

Chacune Région AWS possède plusieurs emplacements isolés appelés zones de disponibilité. Lorsque vous lancez une instance, vous pouvez éventuellement spécifier une zone de disponibilité (AZ) dans la Région AWS que vous utilisez. Flexibilité de la zone de disponibilité est la distribution des instances entre plusieurs zones de disponibilité. Si une instance échoue, vous pouvez concevoir votre application de manière à ce qu'une instance dans une autre AZ puisse traiter les demandes. Pour plus d'informations sur les zones de disponibilité, consultez la documentation sur les régions et zones dans le Guide de l'utilisateur Amazon EC2.

Flexibilité des instances est l'utilisation de plusieurs types d'instances pour répondre aux exigences de capacité. Lorsque vous exprimez de la flexibilité avec les instances, vous pouvez utiliser la capacité globale en fonction de la taille, de la famille et de la génération des instances. Une plus grande flexibilité améliore les chances de trouver et d'allouer la capacité de calcul requise par rapport à un cluster utilisant un seul type d'instance.

La flexibilité des instances et des zones de disponibilité réduit les erreurs de capacité insuffisante (ICE) et les interruptions ponctuelles par rapport à un cluster avec un seul type d'instance ou AZ. Utilisez les bonnes pratiques décrites ici pour déterminer les instances à diversifier une fois que vous connaissez la famille et la taille d'instance initiales. Cette approche maximise la disponibilité des pools de capacité Amazon EC2 avec un minimum de performances et de variations de coûts.

Faire preuve de flexibilité en matière de zones de disponibilité

Nous vous recommandons de configurer toutes les zones de disponibilité à utiliser dans votre cloud privé virtuel (VPC) et de les sélectionner pour votre cluster EMR. Les clusters ne doivent exister que dans une seule zone de disponibilité, mais avec les parcs d'instances Amazon EMR, vous pouvez sélectionner plusieurs sous-réseaux pour 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. Lorsque vous provisionnez un cluster EMR pour plusieurs sous-réseaux, votre cluster peut accéder à un pool de capacités Amazon EC2 plus important que les clusters d'un seul sous-réseau.

Si vous devez prioriser un certain nombre de zones de disponibilité à utiliser dans votre cloud privé virtuel (VPC) pour votre cluster EMR, vous pouvez tirer parti de la fonctionnalité de score de placement ponctuel avec Amazon EC2. Avec le score de placement Spot, vous spécifiez les exigences de calcul pour vos instances Spot, puis EC2 renvoie les dix Régions AWS meilleures zones de disponibilité notées sur une échelle de 1 à 10. Un score de 10 indique que votre demande

Spot a de fortes chances d'aboutir ; un score de 1 indique qu'il est peu probable que votre demande Spot aboutisse. Pour plus d'informations sur l'utilisation de la notation de placement Spot, consultez la section Spot placement score dans le guide de l'utilisateur Amazon EC2.

La flexibilité en ce qui concerne les types d'instances

La flexibilité des instances est l'utilisation de plusieurs types d'instances pour répondre aux exigences de capacité. La fonction de l'instance profite à la fois à l'utilisation d'Amazon EC2 Spot et à la demande. Avec les instances Spot, la flexibilité des instances permet à Amazon EC2 de lancer des instances à partir de pools de capacité plus importants en utilisant des données de capacité en temps réel. Il prédit également les instances les plus disponibles. Cela permet de réduire les interruptions et de réduire le coût global d'une charge de travail. Avec les instances à la demande, la flexibilité des instances réduit les erreurs de capacité insuffisante (ICE) lorsque la capacité totale est allouée à un plus grand nombre de pools d'instances.

Pour les clusters de Groupes d'instances, vous pouvez spécifier jusqu'à 50 types d'instances EC2. Pour les Parcs d'instances dotées d'une stratégie d'allocation, vous pouvez spécifier jusqu'à 30 types d'instances EC2 pour chaque groupe de nœuds principal, principal et de nœuds de tâches. Un plus large éventail d'instances améliore les avantages de la flexibilité des instances.

Exprimer la flexibilité des instances

Tenez compte des bonnes pratiques suivantes pour exprimer la flexibilité des instances pour votre application.

Rubriques

- Déterminer la famille et la taille des instances
- Inclure des instances supplémentaires

Déterminer la famille et la taille des instances

Amazon EMR prend en charge plusieurs types d'instances pour différents cas d'utilisation. Ces types d'instances sont répertoriés dans la documentation Types d’instance pris en charge. Chaque type d'instance appartient à une famille d'instances qui décrit l'application pour laquelle le type est optimisé.

Pour les nouvelles charges de travail, vous devez effectuer une comparaison avec les types d'instances de la famille à usage général, tels que `m5` ou `c5`. Ensuite, surveillez les métriques du système d'exploitation et de YARN à partir de Ganglia et de Amazon CloudWatch pour déterminer

les goulets d'étranglement du système en cas de pic de charge. Les goulots d'étranglement incluent le processeur, la mémoire, le stockage et les opérations d'E/S. Après avoir identifié les goulots d'étranglement, choisissez une solution optimisée pour le calcul, une optimisation pour la mémoire, une optimisation pour le stockage ou une autre famille d'instances adaptée à vos types d'instances. Pour plus de détails, consultez la page Déterminer l'infrastructure adaptée à vos charges de travail Spark dans le guide des meilleures pratiques Amazon EMR sur. GitHub

Identifiez ensuite le plus petit conteneur YARN ou exécuteur Spark dont votre application a besoin. Il s'agit de la plus petite taille d'instance adaptée au conteneur et de la taille d'instance minimale pour le cluster. Utilisez cette métrique pour déterminer les instances avec lesquelles vous pouvez vous diversifier davantage. Une instance plus petite permettra une plus grande flexibilité d'instance.

Pour une flexibilité maximale des instances, vous devez tirer parti du plus grand nombre d'instances possible. Nous vous recommandons de vous diversifier en optant pour des instances présentant des spécifications matérielles similaires. Cela maximise l'accès aux pools de capacité EC2 avec un minimum de variation des coûts et des performances. Diversifiez les tailles. Pour ce faire, donnez d'abord la priorité à AWS Graviton et aux générations précédentes. En règle générale, essayez d'être flexible sur au moins 15 types d'instances pour chaque charge de travail. Nous vous recommandons de commencer par des instances à usage général, optimisées pour le calcul ou des instances à mémoire optimisée. Ces types d'instances offriront la plus grande flexibilité.

Inclure des instances supplémentaires

Pour une diversité maximale, incluez des types d'instances supplémentaires. Priorisez d'abord la taille de l'instance, le graviton et la flexibilité de génération. Cela permet d'accéder à des pools de capacité EC2 supplémentaires présentant des profils de coûts et de performances similaires. Si vous avez besoin d'une flexibilité accrue en raison de l'ICE ou d'interruptions Spot, pensez à la flexibilité des variantes et des familles. Chaque approche comporte des compromis qui dépendent de votre cas d'utilisation et de vos besoins.

- Flexibilité de taille : tout d'abord, diversifiez avec des instances de tailles différentes au sein d'une même famille. Les instances d'une même famille offrent les mêmes coûts et performances, mais peuvent lancer un nombre différent de conteneurs sur chaque hôte. Par exemple, si la taille minimale de l'exécuteur dont vous avez besoin est de 2 vCPU et de 8 Go de mémoire, la taille minimale de l'instance est de `m5.xlarge`. Pour une flexibilité de taille, incluez `m5.xlarge`, `m5.2xlarge`, `m5.4xlarge`, `m5.8xlarge`, `m5.12xlarge`, `m5.16xlarge` et `m5.24xlarge`.
- Flexibilité de Graviton : outre la taille, vous pouvez diversifier grâce aux instances de Graviton. Les instances Graviton sont alimentées par des processeurs AWS Graviton2 qui offrent le

meilleur rapport prix/performances pour les charges de travail dans le cloud sur Amazon EC2. Par exemple, avec une taille d'instance minimale de `m5.xlarge`, vous pouvez inclure `m6g.xlarge`, `m6g.2xlarge`, `m6g.4xlarge`, `m6g.8xlarge`, et `m6g.16xlarge` pour la flexibilité de Graviton.

- Flexibilité de génération : à l'instar de Graviton et de flexibilité de taille, les instances des familles de générations précédentes partagent les mêmes spécifications matérielles. Cela se traduit par un profil de coûts et de performances similaire avec une augmentation du pool Amazon EC2 total accessible. Pour la flexibilité de génération, incluez `m4.xlarge`, `m4.2xlarge`, `m4.10xlarge` et `m4.16xlarge`.
- Flexibilité de famille et de variante
  - Capacité : pour optimiser la capacité, nous recommandons la flexibilité des instances entre les familles d'instances. Les instances communes issues de différentes familles d'instances possèdent des pools d'instances plus profonds qui peuvent aider à répondre aux exigences de capacité. Cependant, les instances de différentes familles auront des ratios vCPU par rapport à la mémoire différents. Cela entraîne une sous-utilisation si le conteneur d'applications attendu est dimensionné pour une instance différente. Par exemple, avec `m5.xlarge`, incluez des instances optimisées pour le calcul, telles que `c5` ou des instances à mémoire optimisée, telles que `r5` pour la flexibilité de famille.
  - Coût : pour optimiser les coûts, nous recommandons la flexibilité des instances entre les variantes. Ces instances ont le même ratio de mémoire et de vCPU que l'instance initiale. L'inconvénient de la flexibilité des variantes est que ces instances ont des pools de capacité plus petits, ce qui peut entraîner une capacité supplémentaire limitée ou un plus grand nombre d'interruptions Spot. Avec `m5.xlarge` par exemple, incluez des instances basées sur AMD (`m5a`), des instances basées sur SSD (`m5d`) ou des instances optimisées pour le réseau (`m5n`) pour la flexibilité des variantes d'instance.

Bonnes pratiques pour la configuration des clusters

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 ?

Il existe plusieurs manières d'ajouter des instances Amazon EC2 à un cluster. La méthode que vous devez choisir varie selon que vous utilisez 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.
  - Configurez le dimensionnement automatique dans Amazon EMR pour un groupe d'instances, en ajoutant et en supprimant des instances automatiquement en fonction de la valeur d'une CloudWatch métrique Amazon que vous spécifiez. Pour plus d'informations, consultez Utiliser la mise à l'échelle des clusters.
- 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 Configuration de parcs d'instances.

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 Affichage et surveillance d'un cluster. 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 primaire, qui attribue les tâches, n'a pas besoin d'une instance EC2 dotée d'une grande puissance de traitement. Les instances Amazon EC2 du type de nœud primaire, 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 Amazon 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 Amazon EC2 disponibles et leur configuration, consultez Configuration des instances Amazon EC2.

Les consignes suivantes s'appliquent à la plupart des clusters Amazon EMR.

- Il existe une limite de vCPU pour le nombre total d'instances Amazon EC2 à la demande que vous exécutez sur AWS un compte par compte. Région AWS Pour plus d'informations sur la limite de vCPU et sur la manière de demander une augmentation de limite pour votre compte, consultez la section Instances à la demande dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

- Le nœud primaire n'a généralement pas de grandes exigences en matière de calcul. Pour les clusters comportant un grand nombre de nœuds, ou pour les clusters dont les applications sont spécifiquement déployées sur le nœud principal (HueJupyterHub, etc.), un nœud principal plus grand peut être nécessaire et peut contribuer à améliorer les performances du cluster. Par exemple, envisagez d'utiliser une instance m5.xlarge pour les petits clusters (50 nœuds ou moins) et de passer à un type d'instance plus grand pour les clusters plus importants.
- 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'instances à usage général, qui offrent des performances équilibrées en termes d'UC, d'espace disque et d'E/S. 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 de calcul en cluster, qui fournissent proportionnellement des ressources d'UC intensives et des performances réseau accrues.
- 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 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.

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
- Nœud primaire sur une instance Spot
- Nœuds de noyau sur les instances Spot
- Nœuds de tâche sur les instances Spot
- Configurations d'instances pour les scénarios d'application

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 tâches, Amazon EMR dispose d'une fonctionnalité par défaut pour planifier les tâches YARN afin que les tâches en cours n'échouent pas lorsque les nœuds de tâches s'exécutant sur des instances Spot sont résiliés. Pour ce faire, Amazon EMR autorise les processus principaux de l'application à s'exécuter uniquement sur les nœuds principaux. Le processus principal de l'application contrôle les tâches en cours d'exécution et doit rester actif pendant toute la durée de vie de la tâche.

Les versions 5.19.0 et ultérieures d'Amazon EMR utilisent la fonctionnalité intégrée d'étiquettes de nœuds YARN 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 afin que le planificateur de capacité YARN et le planificateur équitable tirent parti des étiquettes des nœuds. Amazon EMR étiquette automatiquement les nœuds principaux avec l'étiquette `CORE` et définit les propriétés de manière à ce que les maîtres d'applications soient planifiés uniquement sur les nœuds portant le label CORE. La modification manuelle des propriétés associées dans les classifications de configuration de yarn-site et de capacity-scheduler, ou directement dans les fichiers XML associés, pourrait interrompre cette fonctionnalité ou la modifier.

Amazon EMR configure par défaut les propriétés et valeurs suivantes. Faites attention lorsque vous configurez ces propriétés.

> **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'`

- 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`

Nœud primaire sur une instance Spot

Le nœud primaire contrôle et dirige le cluster. Lorsque le cluster est arrêté, il prend fin, si bien que vous devez lancer uniquement le nœud primaire 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 qu'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 primaire d'instances Spot lorsque vous lancez le cluster. Vous ne pouvez pas ajouter ou supprimer des nœuds primaires d'un cluster en cours d'exécution.

En général, il suffirait d'exécuter le nœud primaire 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 que seules cinq sont disponibles à un prix spot maximal ou inférieur, le groupe d'instances ne sera pas lancé. Amazon EMR continue d'attendre que les six instances Amazon EC2 soient disponibles ou 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”.

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 ou de l'API. AWS CLI 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 du nœud primaire	Option d'achat de nœuds principaux	Option d'achat de nœuds de tâches
Entrepôts des données et clusters de longue durée	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	Spot	Spot	Spot
Charges de travail essentielles pour les données	À la demande	À la demande	Combinaison d'instances Spot ou de parc d'instances
Tests d'application	Spot	Spot	Spot

Il existe plusieurs scénarios dans lesquels les instances Spot sont utiles pour exécuter un cluster Amazon EMR.

Entrepôts des 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 des données), vous pouvez gérer un pic de demande à moindre coût grâce aux instances Spot. Vous pouvez lancer vos groupes d'instances primaires et principales 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 principales, centrales 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 le coût inférieur est plus important que le délai d'exécution, mais que la perte d'un travail partiel n'est pas acceptable, lancez les groupes d'instances principales et de noyau en tant qu'instances à la demande et complétez avec un ou plusieurs groupes d'instances de tâche des instances Spot. L'exécution des groupes d'instances principales et centrales en tant qu'instances à la demande garantit la persistance de vos données dans HDFS et la protection du cluster contre une résiliation en raison des fluctuations du marché Spot, tout en permettant de réaliser des économies qui découlent de l'exécution des groupes d'instances de tâches en tant qu'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 principales, centrales 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 :

- Le nombre d'instances Amazon EC2 utilisées pour les nœuds principaux.
- Capacité du stockage d'instances Amazon EC2 pour le type d'instance utilisé. Pour plus d'informations sur les volumes de stockage d'instance, consultez le magasin d'instances Amazon EC2 dans le guide de l'utilisateur Amazon EC2.
- Nombre et taille des volumes Amazon 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 Amazon 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 Amazon 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 Amazon EBS supplémentaires, ou en ajoutant des groupes d'instances avec des volumes Amazon EBS attachés à un cluster existant
- En ajoutant d'autres nœuds principaux
- En choisissant un type d'instance Amazon 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.

Configuration de la journalisation et du débogage du cluster

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.

Fichiers journaux par défaut

Par défaut, chaque cluster écrit les fichiers journaux sur le nœud primaire. Ils sont écrits dans le répertoire `/mnt/var/log/`. Vous pouvez y accéder à l'aide de SSH pour vous connecter au nœud primaire, comme décrit dans Connexion au nœud primaire à l'aide de SSH.

> **Note**
>
> Si vous utilisez Amazon EMR version 6.8.0 ou antérieure, les fichiers journaux sont enregistrés sur Amazon S3 lors de la résiliation du cluster. Vous ne pouvez donc pas accéder aux fichiers journaux une fois le nœud primaire résilié. Les versions 6.9.0 et ultérieures d'Amazon EMR archivent les journaux sur Amazon S3 pendant la réduction de la taille du cluster, de sorte que les fichiers journaux générés sur le cluster persistent même après que le nœud a été résilié.

Vous n'avez pas besoin d'activer d'options pour que les fichiers journaux soient écrits sur le nœud primaire. 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 :

- Journaux d'étape – Ces journaux sont générés par le service Amazon EMR et contiennent des informations sur le cluster et les résultats de chaque étape. Les fichiers journaux sont stockés dans le répertoire `/mnt/var/log/hadoop/steps/` sur le nœud primaire. Chaque étape enregistre ses résultats dans un sous-répertoire distinct numéroté : `/mnt/var/log/hadoop/steps/s-stepId1/` pour la première étape, `/mnt/var/log/hadoop/steps/s-stepId2/` pour la deuxième étape, et ainsi de suite. Les identifiants d'étape de 13 caractères (par exemple, stepId1, stepId2) sont spécifiques à un cluster.
- Journaux des composants Hadoop et YARN — Les journaux des composants associés à la fois à Apache YARN et MapReduce, par exemple, sont contenus dans des dossiers distincts dans. `/mnt/var/log` Les emplacements des fichiers journaux pour les composants Hadoop dans `/mnt/var/log` sont les suivants : hadoop-hdfs, hadoop-mapreduce, hadoop-httpfs et hadoop-yarn. Le hadoop-state-pusher répertoire est destiné à la sortie du processus Hadoop State Pusher.
- Les journaux des actions d'amorçage – Si votre travail utilise des actions d'amorçage, les résultats de ces actions sont enregistrés. Les fichiers journaux sont stockés dans /mnt/var/log/bootstrap-actions/ sur le nœud primaire. Chaque étape enregistre ses résultats dans un sous-répertoire distinct numéroté : `/mnt/var/log/bootstrap-actions/1/` pour la première action

d'amorçage, `/mnt/var/log/bootstrap-actions/2/` pour la deuxième action d'amorçage, et ainsi de suite.

- Les journaux d'état de l'instance – Ces journaux fournissent des informations sur l'UC, l'état de la mémoire et les threads de nettoyage de mémoire du nœud. Les fichiers journaux sont stockés dans `/mnt/var/log/instance-state/` sur le nœud primaire.

Archiver les fichiers journaux sur Amazon S3

> **Note**
>
> Vous ne pouvez pas actuellement utiliser l'agrégation des journaux vers Amazon S3 avec l'utilitaire `yarn logs`.

Les versions 6.9.0 et ultérieures d'Amazon EMR archivent les journaux sur Amazon S3 pendant la réduction de la taille du cluster, de sorte que les fichiers journaux générés sur le cluster persistent même après que le nœud a été résilié. Ce comportement étant activé automatiquement, vous n'avez rien à faire pour l'activer. Pour les versions 6.8.0 et antérieures d'Amazon EMR, vous pouvez configurer un cluster pour archiver régulièrement les fichiers journaux stockés sur le nœud primaire vers Amazon S3. Vous avez ainsi la garantie que les fichiers journaux sont disponibles une fois que le cluster est résilié, qu'il s'agisse d'une fermeture normale ou d'une erreur. Amazon EMR archive les fichiers journaux sur Amazon S3 toutes les 5 minutes.

Pour que les fichiers journaux soient archivés dans Amazon S3 pour Amazon EMR versions 6.8.0 et antérieures, 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.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour archiver les fichiers journaux sur Amazon S3 avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Journaux du cluster, cochez la case Publier les journaux spécifiques au cluster sur Amazon S3.
4. Dans le champ Emplacement Amazon S3, saisissez (ou naviguez jusqu'à) un chemin Amazon S3 pour stocker vos journaux. Si vous tapez le nom d'un dossier qui n'existe pas dans le compartiment S3, Amazon S3 le crée.

   Lorsque vous définissez cette valeur, Amazon EMR copie les fichiers journaux des instances EC2 du cluster sur Amazon S3. Cela évite que les fichiers journaux ne soient perdus lorsque le cluster se termine et que l'EC2 résilie les instances hébergeant le cluster. Ces journaux sont utiles à des fins de dépannage. Pour plus d'informations, consultez Affichage des fichiers journaux.
5. Cochez éventuellement la case Chiffrer les journaux spécifiques au cluster. Sélectionnez ensuite une AWS KMS clé dans la liste, entrez un ARN de clé ou créez une nouvelle clé. Cette option n'est disponible qu'avec Amazon EMR version 5.30.0 et ultérieure, à l'exclusion de la version 6.0.0. Pour utiliser cette option, ajoutez une autorisation AWS KMS pour votre profil d'instance EC2 et votre rôle Amazon EMR. Pour plus d’informations, consultez Pour chiffrer les fichiers journaux stockés dans Amazon S3 avec une clé AWS KMS gérée par le client.
6. Choisissez toutes les autres options qui s'appliquent à votre cluster.
7. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour archiver les fichiers journaux sur Amazon S3 avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.

2. Choisissez Créer un cluster.
3. Choisissez Accéder aux options avancées.
4. Dans la section Options générales, dans le champ Journalisation, acceptez l'option par défaut : Activé.

   Cette option détermine si Amazon EMR capture des données de journal détaillées sur Amazon S3. Vous ne pouvez définir cette option que lors de la création du cluster. Pour plus d'informations, consultez Afficher les fichiers journaux .
5. Dans le champ Dossier S3, saisissez (ou naviguez jusqu'à) un chemin vers 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 plus d'informations, consultez Affichage des 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 AWS KMS clé.

   Cette option n'est disponible qu'avec Amazon EMR version 5.30.0 et ultérieure, à l'exclusion de la version 6.0.0. Pour utiliser cette option, ajoutez l'autorisation à AWS KMS pour votre profil d'instance EC2 et votre rôle Amazon EMR. Pour plus d'informations, consultez Pour chiffrer les fichiers journaux stockés dans Amazon S3 avec une clé AWS KMS gérée par le client.
7. Procédez à la création du cluster, comme décrit dans Planification et configuration des clusters.

CLI

Pour archiver des fichiers journaux sur Amazon S3 à l'aide du AWS CLI

Pour archiver les fichiers journaux sur Amazon S3 à l'aide du AWS CLI, tapez la `create-cluster` commande et spécifiez le chemin du journal Amazon S3 à l'aide du `--log-uri` paramètre.

1. Pour enregistrer des fichiers dans Amazon S3 tapez la commande suivante et remplacez *myKey* par le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.1.0 --log-
uri s3://DOC-EXAMPLE-BUCKET/logs --applications Name=Hadoop Name=Hive Name=Pig
 --use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --
instance-count 3
```

2. Lorsque vous spécifiez le nombre d'instances sans utiliser le paramètre `--instance-groups`, un seul nœud primaire 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, saisissez `aws emr create-default-roles` pour les créer avant de taper la sous-commande `create-cluster`.

Pour chiffrer les fichiers journaux stockés dans Amazon S3 avec une clé AWS KMS gérée par le client

Avec Amazon EMR version 5.30.0 et versions ultérieures (sauf Amazon EMR 6.0.0), vous pouvez chiffrer les fichiers journaux stockés dans Amazon S3 à l'aide d'une clé gérée par le client KMS. AWS Pour activer cette option dans la console, suivez les étapes de la section Archiver les fichiers journaux sur Amazon S3. Votre profil d'instance Amazon EC2 et votre rôle Amazon EMR doivent répondre aux conditions préalables suivantes :

- Le profil d'instance Amazon EC2 utilisé pour votre cluster doit être autorisé à utiliser `kms:GenerateDataKey`.
- Le rôle Amazon EMR utilisé pour votre cluster doit avoir l'autorisation d'utiliser `kms:DescribeKey`.

- Le profil d'instance Amazon EC2 et le rôle Amazon EMR doivent être ajoutés à la liste des utilisateurs clés pour la clé gérée par le client AWS KMS spécifiée, comme le montrent les étapes suivantes :

  1. Ouvrez la console AWS Key Management Service (AWS KMS) à l'adresse https://console.aws.amazon.com/kms.
  2. Pour modifier la AWS région, utilisez le sélecteur de région dans le coin supérieur droit de la page.
  3. Sélectionnez l'alias de la clé KMS 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 votre profil d'instance Amazon EC2 et votre rôle Amazon EMR.
  6. Choisissez Ajouter.

Pour plus d'informations, consultez les rôles de service IAM utilisés par Amazon EMR et l'utilisation de politiques clés dans AWS le guide du développeur du service de gestion des clés.

Pour regrouper les journaux dans Amazon S3 à l'aide du fichier de configuration 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 vers Amazon S3 à l'aide de AWS CLI, vous devez utiliser une action bootstrap au lancement du cluster afin d'activer l'agrégation des journaux et de spécifier le compartiment dans lequel les journaux seront stockés.

- Pour activer le regroupement des journaux, créez le fichier de configuration suivant, appelé `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:\/\/DOC-EXAMPLE-BUCKET\/logs"
    }
  }
]
```

Tapez la commande suivante et remplacez *myKey* par le nom de votre paire de clés EC2. Vous pouvez également remplacer n'importe quel texte rouge par vos propres configurations.

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.1.0 \
--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 primaire 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 et le profil d'instance EC2 par défaut, exécutez `aws emr create-default-roles` pour les créer avant d'exécuter la sous-commande `create-cluster`.

Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le AWS CLI, consultez AWS CLI Command Reference.

Emplacements des journaux

La liste suivante inclut tous les types de journaux et leur emplacement dans Amazon S3. Vous pouvez les utiliser pour résoudre les problèmes liés à Amazon EMR.

Journaux d'étapes

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/steps/<step-id>/`

Journaux d'application

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/containers/`

Cet emplacement inclut le conteneur `stderr` et `stdout`, `directory.info`, `prelaunch.out`, et les journaux `launch_container.sh`.

Journaux du gestionnaire de ressources

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/applications/hadoop-yarn/`

Hadoop HDFS

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/applications/hadoop-hdfs/`

Cet emplacement inclut NameNode DataNode, et les TimelineServer journaux YARN.

Journaux du gestionnaire de nœuds

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/applications/hadoop-yarn/`

Journaux d'état de l'instance

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/daemons/instance-state/`

Journaux de provisionnement d'Amazon EMR

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/provision-node/*`

Journaux de la ruche

`s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/applications/hive/*`

- Pour trouver les journaux Hive sur votre cluster, supprimez l'astérisque (*) et ajoutez `/var/log/hive/` au lien ci-dessus.
- Pour trouver HiveServer 2 journaux, supprimez l'astérisque (*) et ajoutez-les `var/log/hive/hiveserver2.log` au lien ci-dessus.

- Pour trouver les journaux HiveCLI, supprimez l'astérisque (*) et ajoutez `/var/log/hive/user/hadoop/hive.log` au lien ci-dessus.
- Pour trouver les journaux Hive Metastore Server, supprimez l'astérisque (*) et ajoutez `/var/log/hive/user/hive/hive.log` au lien ci-dessus.

Si votre échec se situe dans le nœud principal ou le nœud de tâche de votre application Tez, fournissez les journaux du conteneur Hadoop approprié.

Activation de l'outil de débogage

L'outil de débogage vous permet d'explorer plus facilement les fichiers journaux depuis la console Amazon EMR. Pour plus d'informations, consultez Afficher des fichiers journaux dans l'outil de débogage. 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 Amazon 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. Notez que la nouvelle console Amazon EMR ne propose pas d'outil de débogage.

Old console

Pour activer l'outil de débogage avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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 Emplacement S3 du dossier des journaux, saisissez un chemin 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 la page produit Amazon SQS.

7. Procédez à la création du cluster, comme décrit dans Planification et configuration des clusters.

AWS CLI

Pour activer l'outil de débogage à l'aide du AWS CLI

Pour activer le débogage à l'aide de AWS CLI, tapez la `create-cluster` sous-commande avec le `--enable-debugging` paramètre. Vous devez également spécifier le paramètre `--log-uri` lorsque vous activez le débogage.

- Pour activer le débogage à l'aide de AWS CLI, tapez la commande suivante et remplacez *MyKey* par le nom de votre paire de clés EC2.

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.1.0 \
--log-uri s3://DOC-EXAMPLE-BUCKET/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 primaire 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`.

API

Pour activer l'outil de débogage avec l'API Amazon EMR

- Activez le débogage à l'aide de la configuration du SDK Java suivante.

```
   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.elasticmapreduce")`, par exemple `new StepFactory("ap-northeast-2.elasticmapreduce")`.

Informations relatives aux options de débogage

Les versions 4.1.0 à 5.27.0 d'Amazon EMR prennent en charge le débogage dans toutes les régions. Les autres versions d'Amazon EMR ne prennent pas en charge l'option de débogage. À compter du 23 janvier 2023, Amazon EMR arrêtera l'outil de débogage pour toutes les versions.

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 https://aws.amazon.com/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 Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..

Clusters de balise

Il peut être pratique de classer vos AWS ressources de différentes manières, par exemple par objectif, propriétaire ou environnement. Vous pouvez effectuer cela dans Amazon EMR en

affectant des métadonnées personnalisées pour vos clusters Amazon EMR à l'aide de balises. une identification est constituée d'une clé et d'une valeur que vous définissez. Pour Amazon EMR, le cluster est le niveau de ressource 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 propagée 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 de groupe d'instance et *j-12345678* est un exemple de valeur d'identifiant d'un 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 travailler avec des balises à l' AWS Management Console aide de la CLI 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 obtenir des autorisations basées sur les ressources par balise, vos politiques IAM sont appliquées aux balises qu'Amazon EMR propage aux instances Amazon EC2 d'un cluster. Pour que les balises Amazon EMR puissent se propager à vos instances Amazon EC2, votre politique IAM pour Amazon EC2 doit autoriser les appels à Amazon EC2 et aux API. CreateTags 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 politique IAM, tout comme les autres balises Amazon EC2. Gardez votre politique IAM à l'esprit lorsque vous ajoutez des balises à vos clusters Amazon EMR afin d'éviter qu'un utilisateur ne dispose d'autorisations incorrectes pour un cluster. Pour éviter les problèmes, assurez-vous que vos politiques IAM n'incluent pas de conditions sur les balises que vous prévoyez également d'utiliser sur vos clusters Amazon EMR. Pour plus d'informations, consultez Contrôle de l'accès aux ressources Amazon EC2.

Restrictions liées aux étiquettes

Les restrictions de base suivantes s’appliquent aux balises :

- Les restrictions qui s'appliquent aux ressources Amazon EC2 s'appliquent également à Amazon EMR. Pour plus d’informations, consultez https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions.
- N'utilisez pas le `aws:` préfixe dans les noms et les valeurs des balises car il est réservé à AWS l'usage. De même, vous ne pouvez pas modifier ni supprimer des noms ou valeurs de balise ayant ce préfixe.
- Vous ne pouvez pas changer ni modifier de balises sur un cluster hors service.
- Une valeur de balise peut être une chaîne vide, mais pas null. En outre, une clé de balise ne peut pas être une chaîne vide.

- Les clés et les valeurs peuvent contenir tout caractère alphabétique dans n'importe quelle langue, tout caractère numérique, espace blanc, séparateur invisible et les symboles suivants : _ . : / = + - @

Pour plus d'informations sur le balisage à l'aide du AWS Management Console, consultez la section Utilisation des balises dans la console dans le guide de l'utilisateur Amazon EC2. Pour plus d'informations sur le balisage à l'aide de l'API Amazon EC2 ou de la ligne de commande, consultez la présentation de l'API et de la CLI dans le guide de l'utilisateur Amazon EC2.

Balisage des ressources pour la facturation

Vous pouvez utiliser des balises pour organiser votre AWS facture afin de refléter votre propre structure de coûts. Pour ce faire, inscrivez-vous pour obtenir la facture de votre AWS compte avec les valeurs clés du tag 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 plus d'informations, consultez la section Allocation des coûts et balisage dans le Guide de l'utilisateur de AWS Billing .

Ajout de balises à un cluster

Vous pouvez ajouter des balises à un cluster lorsque vous le créez.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour ajouter des balises lorsque vous créez un cluster avec la nouvelle console.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Balises, choisissez Ajouter une nouvelle balise. Spécifiez une balise dans le champ Clé. Vous pouvez également spécifier une balise dans le champ Valeur.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour ajouter des balises lorsque vous créez un cluster avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster et Go to advanced options (Aller aux options avancées).
3. Sur la page Step 3: General Cluster Settings (Étape 3 : Paramètres généraux de cluster) dans la section Balises, saisissez une Clé pour votre balise.

   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.

AWS CLI

Pour ajouter des balises lorsque vous créez un cluster à l'aide du 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 de clé *`marketing`* lors de la création d'un cluster, tapez la commande suivante et remplacez *`myKey`* par 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Vous pouvez également ajouter des balises à un cluster existant.

New console

Pour ajouter des balises à un cluster existant avec la nouvelle console

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

2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Balises de la page de détails du cluster, sélectionnez Gérer les balises. Spécifiez une balise dans le champ Clé. Vous pouvez également spécifier une balise dans le champ Valeur.
4. Sélectionnez Enregistrer les modifications. L'onglet Balises est mis à jour avec le nouveau nombre de balises que vous avez sur votre cluster. Par exemple, si vous avez maintenant deux balises, la balise de votre onglet est Balises (2).

Old console

Pour ajouter des balises à un cluster existant avec l'ancienne 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.

AWS CLI

Pour ajouter des balises à un cluster en cours d'exécution à l'aide du AWS CLI

- Entrez la sous-commande `add-tags` avec le paramètre `--tag` pour attribuer des balises à l'ID du cluster. Vous pouvez trouver l'ID du cluster en utilisant la console ou la commande `list-clusters`. Actuellement, la sous-commande `add-tags` n'accepte qu'un ID de ressource.

  Par exemple, pour ajouter deux balises à un cluster en cours d'exécution, l'une avec une clé nommée *`costCenter`* avec la valeur *`marketing`* et l'autre nommée *`other`* avec la valeur

accounting, entrez la commande suivante et remplacez *j-KT4XXXXXXXX1NM* par l'ID de votre cluster.

```
aws emr add-tags --resource-id j-KT4XXXXXXXX1NM --tag "costCenter=marketing" --
tag "other=accounting"
```

Notez que lorsque des balises sont ajoutées à l'aide de la AWS CLI, la commande ne produit aucun résultat. Pour plus d'informations sur l'utilisation des commandes Amazon EMR dans le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Affichage des balises sur un cluster

Si vous voulez voir toutes les balises associées à un cluster, vous pouvez les afficher avec la console ou l' AWS CLI.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour afficher les balises d'un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Pour afficher tous vos balises, sélectionnez l'onglet Balises sur la page des détails du cluster.

Old console

Pour afficher les balises d'un cluster avec l'ancienne 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.

AWS CLI

Pour afficher les balises d'un cluster avec AWS CLI

Pour afficher les balises d'un cluster à l'aide de AWS CLI, tapez la `describe-cluster` sous-commande avec le `--query` paramètre.

- Pour afficher les balises d'un cluster, tapez la commande suivante et remplacez *j-KT4XXXXXXXX1NM* par l'ID de votre 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Retrait des balises d'un cluster

Si vous n'avez plus besoin une balise, vous pouvez la supprimer du cluster.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour supprimer les balises d'un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Balises de la page de détails du cluster, sélectionnez Gérer les balises.
4. Choisissez Supprimer pour chaque paire clé-valeur que vous souhaitez supprimer.
5. Sélectionnez Enregistrer les modifications.

Old console

Pour supprimer des balises sur un cluster à l'aide de l'ancienne 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.

AWS CLI

Pour supprimer des balises sur un cluster à l'aide du 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* par l'ID de votre 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Intégration de pilotes et d'applications tierces

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

- Utilisation d'outils de business intelligence avec Amazon EMR

Utilisation d'outils de business intelligence avec Amazon EMR

Vous pouvez utiliser des outils de business intelligence populaires tels que Microsoft Excel et Tableau avec Amazon EMR pour explorer et visualiser vos données. MicroStrategy QlikView Un grand nombre de ces outils ont besoin d'un pilote ODBC (Open Database Connectivity) ou JDBC (Java Database Connectivity). Pour télécharger et installer les derniers pilotes, consultez http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/.

Pour trouver les anciennes versions des pilotes, consultez http://awssupportdatasvcs.com/bootstrap-actions/Simba/.

Sécurité dans Amazon EMR

La sécurité et la conformité sont une responsabilité que vous partagez avec vous AWS. Ce modèle de responsabilité partagée peut vous aider à alléger votre charge opérationnelle en AWS exploitant, en gérant et en contrôlant les composants, depuis le système d'exploitation hôte et la couche de virtualisation jusqu'à la sécurité physique des installations dans lesquelles les clusters EMR opèrent. Vous assumez la responsabilité, la gestion et la mise à jour des clusters Amazon EMR, ainsi que la configuration du logiciel d'application et des contrôles de sécurité AWS fournis. Cette différenciation des responsabilités est communément appelée sécurité du cloud par rapport à la sécurité dans le cloud.

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui s'y exécute Services AWS AWS. AWS vous fournit également des 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 conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon EMR, consultez Services AWS la section Champ d'application par programme de conformité.
- Sécurité dans le cloud : vous êtes également chargé d'effectuer toutes les tâches de configuration et de gestion de sécurité nécessaires pour sécuriser un cluster Amazon EMR. Les clients qui déploient un cluster Amazon EMR sont responsables de la gestion du logiciel d'application installé sur les instances et de la configuration des fonctionnalités AWS fournies, telles que les groupes de sécurité, le chiffrement et le contrôle d'accès, conformément à vos exigences, aux lois et réglementations applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon EMR. Les rubriques de ce chapitre vous montrent comment configurer Amazon EMR et en utiliser d'autres Services AWS pour atteindre vos objectifs de sécurité et de conformité.

Sécurité du réseau et de l'infrastructure

En tant que service géré, Amazon EMR est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité. AWS les services de protection du réseau et de l'infrastructure vous offrent des protections précises à la fois au niveau de l'hôte et au niveau du réseau. Amazon EMR prend en charge Services AWS et

propose des fonctionnalités d'application qui répondent aux exigences de conformité et de protection de votre réseau.

- Les groupes de sécurité Amazon EC2 agissent comme un pare-feu virtuel pour les instances de cluster Amazon EMR, limitant le trafic réseau entrant et sortant. Pour plus d'informations, consultez la section Contrôler le trafic réseau à l'aide de groupes de sécurité.
- Amazon EMR Block Public Access (BPA) vous empêche de lancer un cluster dans un sous-réseau public si le cluster possède une configuration de sécurité qui autorise le trafic entrant depuis des adresses IP publiques sur un port. Pour plus d'informations, consultez Utiliser Amazon EMR pour bloquer l'accès public.
- Secure Shell (SSH) permet aux utilisateurs de se connecter en toute sécurité à la ligne de commande sur les instances de cluster. Vous pouvez également utiliser SSH pour afficher les interfaces Web hébergées par les applications sur le nœud principal d'un cluster. Pour plus d'informations, consultez Utiliser une paire de clés EC2 pour les informations d'identification SSH et Connect to a cluster.

Mises à jour de l'AMI Amazon Linux par défaut pour Amazon EMR

> **Important**
>
> Les clusters Amazon EMR qui exécutent des AMI (Amazon Machine Images) Amazon Linux ou Amazon Linux 2 utilisent le comportement par défaut d'Amazon Linux et ne téléchargent pas et n'installent pas automatiquement les mises à jour importantes et critiques du noyau nécessitant un redémarrage. Ce comportement est identique à celui des autres instances Amazon EC2 qui exécutent l'AMI Amazon Linux par défaut. Si de nouvelles mises à jour logicielles Amazon Linux nécessitant un redémarrage (telles que les mises à jour du noyau, de NVIDIA et de CUDA) sont disponibles après la publication d'une version d'Amazon EMR, les instances de cluster EMR qui exécutent l'AMI par défaut ne téléchargent pas et n'installent pas automatiquement ces mises à jour. Pour obtenir les mises à jour du noyau, vous pouvez personnaliser votre AMI Amazon EMR afin d'utiliser la dernière AMI Amazon Linux.

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 plus d'informations, consultez Utilisation de l'AMI Amazon Linux par défaut pour Amazon EMR. Notez que votre configuration réseau doit autoriser la sortie HTTP et HTTPS vers les référentiels Linux d'Amazon S3, sinon les mises à jour de sécurité échoueront.

AWS Identity and Access Management avec Amazon EMR

AWS Identity and Access Management (IAM) est un AWS service qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs IAM contrôlent les personnes qui peuvent être authentifiées (connectées) et autorisées (disposant d'autorisations) à utiliser des ressources Amazon EMR. Les identités IAM incluent les utilisateurs, les groupes et les rôles. Un rôle IAM est similaire à un utilisateur IAM, mais il n'est pas associé à une personne spécifique et est destiné à être assumé par tout utilisateur ayant besoin d'autorisations. Pour plus d'informations, consultez AWS Identity and Access Management Amazon EMR. Amazon EMR utilise plusieurs rôles IAM pour vous aider à mettre en œuvre des contrôles d'accès pour les clusters Amazon EMR. IAM est un AWS service que vous pouvez utiliser sans frais supplémentaires.

- Rôle IAM pour Amazon EMR (rôle EMR) : contrôle la manière dont le service Amazon EMR peut accéder à d' Services AWS autres personnes en votre nom, par exemple en fournissant des instances Amazon EC2 lors du lancement du cluster Amazon EMR. Pour plus d'informations, consultez Configurer les rôles de service IAM pour les autorisations Services AWS et les ressources Amazon EMR.
- Rôle IAM pour les instances EC2 du cluster (profil d'instance EC2) : rôle attribué à chaque instance EC2 du cluster Amazon EMR lors du lancement de l'instance. Les processus d'application qui s'exécutent sur le cluster utilisent ce rôle pour interagir avec d'autres processus Services AWS, tels qu'Amazon S3. Pour plus d'informations, consultez la section Rôle IAM pour les instances EC2 du cluster.
- Rôle IAM pour les applications (rôle d'exécution) : rôle IAM que vous pouvez spécifier lorsque vous soumettez une tâche ou une requête à un cluster Amazon EMR. La tâche ou la requête que vous soumettez à votre cluster Amazon EMR utilise le rôle d'exécution pour accéder à AWS des ressources, telles que des objets dans Amazon S3. Vous pouvez spécifier des rôles d'exécution avec Amazon EMR pour les tâches Spark et Hive. En utilisant les rôles d'exécution, vous pouvez isoler les tâches exécutées sur le même cluster en utilisant différents rôles IAM. Pour plus d'informations, consultez Utilisation du rôle IAM comme rôle d'exécution avec Amazon EMR.

Les identités du personnel font référence aux utilisateurs qui créent ou exploitent des charges de travail. AWS Amazon EMR fournit un support pour les identités du personnel avec les éléments suivants :

- AWS Le centre d'identité IAM (Idc) est recommandé Service AWS pour gérer l'accès des utilisateurs aux AWS ressources. Il s'agit d'un endroit unique où vous pouvez attribuer les identités de vos employés, ainsi qu'un accès cohérent à plusieurs AWS comptes et applications. Amazon EMR prend en charge l'identité des employés grâce à une propagation d'identité fiable. Grâce à la fonctionnalité de propagation d'identité fiable, un utilisateur peut se connecter à l'application et cette application peut transmettre l'identité de l'utilisateur à d'autres personnes Services AWS pour autoriser l'accès aux données ou aux ressources. Pour plus d'informations, consultez Activer la prise en charge du centre d'identitéAWS IAM avec Amazon EMR.

  Le protocole LDAP (Lightweight Directory Access Protocol) est un protocole d'application standard ouvert, indépendant du fournisseur et destiné à accéder aux informations relatives aux utilisateurs, aux systèmes, aux services et aux applications sur le réseau et à les gérer. LDAP est couramment utilisé pour l'authentification des utilisateurs sur des serveurs d'identité d'entreprise tels qu'Active Directory (AD) et OpenLDAP. En activant LDAP avec les clusters EMR, vous permettez aux utilisateurs d'utiliser leurs informations d'identification existantes pour s'authentifier et accéder aux clusters. Pour plus d'informations, consultez la section Activation de la prise en charge du protocole LDAP avec Amazon EMR.

  Kerberos est un protocole d'authentification réseau conçu pour fournir une authentification forte aux applications client/serveur à l'aide de la cryptographie à clé secrète. Lorsque vous utilisez Kerberos, Amazon EMR configure Kerberos pour les applications, les composants et les sous-systèmes qu'il installe sur le cluster afin qu'ils soient authentifiés les uns avec les autres. Pour accéder à un cluster avec Kerberos configuré, un Kerberos principal doit être présent dans le contrôleur de domaine Kerberos (KDC). Pour plus d'informations, consultez Activation de la prise en charge de Kerberos avec Amazon EMR.

Clusters à locataire unique et à locataires multiples

Un cluster est configuré par défaut pour une location unique avec le profil d'instance EC2 comme identité IAM. Dans un cluster à locataire unique, chaque tâche dispose d'un accès complet au cluster et l'accès à toutes les Services AWS ressources est effectué sur la base du profil d'instance EC2. Dans un cluster à locataires multiples, les locataires sont isolés les uns des autres et n'ont pas un accès complet aux clusters et aux instances EC2 du cluster. L'identité sur les clusters à locataires

multiples est soit les rôles d'exécution, soit les identifiants du personnel. Dans un cluster mutualisé, vous pouvez également activer la prise en charge du contrôle d'accès détaillé (FGAC) via AWS Lake Formation Apache Ranger. Dans le cas d'un cluster dont les rôles d'exécution ou le FGAC sont activés, l'accès au profil d'instance EC2 est également désactivé via iptables.

> **Important**
>
> Tous les utilisateurs ayant accès à un cluster à locataire unique peuvent installer n'importe quel logiciel sur le système d'exploitation (OS) Linux, modifier ou supprimer des composants logiciels installés par Amazon EMR et avoir un impact sur les instances EC2 qui font partie du cluster. Si vous souhaitez vous assurer que les utilisateurs ne peuvent pas installer ou modifier les configurations d'un cluster Amazon EMR, nous vous recommandons d'activer la mutualisation pour le cluster. Vous pouvez activer la mutualisation sur un cluster en activant la prise en charge du rôle d'exécution, du centre d'identité AWS IAM, de Kerberos ou du LDAP.

Protection des données

Avec AWS, vous contrôlez vos données en utilisant Services AWS des outils pour déterminer comment les données sont sécurisées et qui y a accès. Des services tels que AWS Identity and Access Management (IAM) vous permettent de gérer en toute sécurité l'accès Services AWS et les ressources. AWS CloudTrail permet la détection et l'audit. Amazon EMR vous permet de chiffrer facilement les données au repos dans Amazon S3 à l'aide de clés que vous gérez AWS ou que vous gérez entièrement. Amazon EMR prend également en charge l'activation du chiffrement des données en transit. Pour plus d'informations, consultez la section Chiffrer les données au repos et en transit.

Contrôle d'accès aux données

Grâce au contrôle d'accès aux données, vous pouvez contrôler les données auxquelles une identité IAM ou une identité de personnel peut accéder. Amazon EMR prend en charge les contrôles d'accès suivants :

- Politiques basées sur l'identité IAM : gérez les autorisations pour les rôles IAM que vous utilisez avec Amazon EMR. Les politiques IAM peuvent être combinées avec le balisage pour contrôler l'accès sur une cluster-by-cluster base. Pour plus d'informations, consultez AWS Identity and Access Management Amazon EMR.

- AWS Lake Formationcentralise la gestion des autorisations de vos données et facilite leur partage au sein de votre organisation et en externe. Vous pouvez utiliser Lake Formation pour permettre un accès précis au niveau des colonnes aux bases de données et aux tables du Glue AWS Data Catalog. Pour plus d'informations, consultez la section Utilisation AWS Lake Formation avec Amazon EMR.
- L'accès Amazon S3 accorde des identités de mappage aux identités de mappage figurant dans des annuaires tels qu'Active Directory, ou AWS Identity and Access Management (IAM) principals, et à des ensembles de données dans S3. En outre, l'accès S3 accorde l'identité de l'utilisateur final du journal et l'application utilisée pour accéder aux AWS CloudTrail données S3. Pour plus d'informations, consultez Utilisation des autorisations d'accès Amazon S3 avec Amazon EMR.
- Apache Ranger est un framework permettant d'activer, de surveiller et de gérer la sécurité complète des données sur la plateforme Hadoop. Amazon EMR prend en charge le contrôle d'accès précis basé sur Apache Ranger pour Apache Hive Metastore et Amazon S3. Pour plus d'informations, consultez Intégrer Apache Ranger à Amazon EMR.

Utilisation de configurations de sécurité pour configurer la sécurité du cluster

Vous pouvez utiliser les configurations de sécurité Amazon EMR pour configurer le chiffrement des données, l’authentification Kerberos et l’autorisation Amazon S3 pour EMRFS sur vos clusters. Tout d’abord, vous créez une configuration de sécurité. Cette configuration peut ensuite être utilisée et réutilisée lorsque vous créez des clusters.

Vous pouvez utiliser le AWS Management Console, le AWS Command Line Interface (AWS CLI) ou les AWS SDK pour créer des configurations de sécurité. Vous pouvez également utiliser un AWS CloudFormation modèle pour créer une configuration de sécurité. Pour plus d'informations, consultez le Guide de AWS CloudFormation l'utilisateur et le modèle de référence pour AWS::EMR::SecurityConfiguration.

Rubriques

- Création d'une configuration de sécurité
- Spécification d'une configuration de sécurité pour un cluster

Création d'une configuration de sécurité

Cette rubrique décrit les procédures générales permettant de créer une configuration de sécurité avec la console Amazon EMR et le AWS CLI, suivie d'une référence pour les paramètres relatifs au chiffrement, à l'authentification et aux rôles IAM pour EMRFS. Pour plus d'informations sur ces fonctions, consultez les rubriques suivantes :

- Chiffrer les données au repos et en transit
- Utilisation de Kerberos pour l'authentification avec Amazon EMR
- Configuration de rôles IAM pour les demandes EMRFS à Amazon S3

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/emr.
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. Choisissez les options pour Chiffrement et Authentification comme décrit dans les sections ci-dessous, puis cliquez sur Créer.

Pour créer une configuration de sécurité à l'aide du AWS CLI

- Utilisez la commande `create-security-configuration`, comme illustré dans l'exemple suivant.
  - Dans *`SecConfigNom`*, 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 d'accès à un fichier JSON local, comme *`file://MySecConfig.json`*. Les paramètres JSON définissent les options pour le Chiffrement, les Rôles IAM pour l'accès EMRFS à Amazon S3 et l'Authentification, comme décrit dans les sections suivantes.

```
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 plus d'informations, consultez Fournir les clés de chiffrement des données au repos avec Amazon EMR et Fournir des certificats de chiffrement des données en transit avec le chiffrement Amazon EMR.

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 des 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.

> **Note**
>
> Lorsque vous les utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour plus d'informations, consultez Tarification d'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 Chiffrement des données S3, pour 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 AWS KMS gérées (SSE-KMS) ou le chiffrement côté client avec des clés gérées (CSE-KMS). AWS KMS Pour AWS KMS key, sélectionnez une clé. Cette clé doit être dans la même région que votre cluster EMR. Pour voir les exigences relatives aux clés, consultez Utilisation AWS KMS keys pour le chiffrement.

- CSE-Custom

  Spécifie le chiffrement côté client via une clé racine personnalisée côté client (CSE-custom). Pour Objet S3, saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Ensuite, pour la classe du fournisseur de clés, entrez le nom complet de la classe déclarée dans votre application qui implémente l'EncryptionMaterialsProvider interface.

- Sous Local disk encryption (Chiffrement de disque local), choisissez une valeur pour Key provider type (Type de fournisseur de clé).

  - AWS KMS key

    Sélectionnez cette option pour spécifier une AWS KMS key. Pour AWS KMS key, 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 AWS KMS keys pour le chiffrement.

    Chiffrement EBS

    Lorsque vous AWS KMS le spécifiez comme fournisseur de clés, vous pouvez activer le chiffrement EBS pour chiffrer le périphérique racine EBS et les volumes de stockage. Pour activer cette option, vous devez accorder à la fonction du service Amazon EMR `EMR_DefaultRole` les autorisations nécessaires pour utiliser la AWS KMS key 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 KMS.

  - Personnalisé

    Sélectionnez cette option pour spécifier un fournisseur de clés personnalisé. Pour Objet S3, saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Pour la classe du fournisseur de clés, entrez le nom complet de la classe déclarée dans votre application qui implémente l' EncryptionMaterialsProvider interface. 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 pour plus de détails. Pour 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 Objet S3, saisissez l'emplacement dans Amazon S3 ou l'ARN Amazon S3 du fichier JAR de votre fournisseur de clés personnalisé. Pour la classe du fournisseur de clés, entrez le nom complet de la classe déclarée dans votre application qui implémente l'ArtifactsProvider interface TLS.

Spécification des options de chiffrement à l'aide du 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 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 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 du disque local est utilisé 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 du disque local est utilisé 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"
   }
  }
 }
}'
```

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.
- Plusieurs AWS KMS clés sont utilisées, une par compartiment S3, et des exceptions de chiffrement sont appliquées à ces 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 du disque local est activé en AWS KMS tant que 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 du disque local est activé en AWS KMS tant que 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"
            }
        }
     }
}'
```

L'exemple suivant illustre le scénario suivant :

Le SSE-EMR-WAL est utilisé pour le chiffrement EMR WAL

```
aws emr create-security-configuration --name "MySecConfig" \
    --security-configuration '{
        "EncryptionConfiguration": {
            "EMRWALEncryptionConfiguration":{ },
            "EnableInTransitEncryption":false, "EnableAtRestEncryption":false
        }
    }'
```

`EnableInTransitEncryption`net pourrait `EnableAtRestEncryption` toujours être vrai, si vous souhaitez activer le chiffrement associé.

L'exemple suivant illustre le scénario suivant :

- Le SSE-KMS-WAL est utilisé pour le chiffrement EMR WAL
- Le chiffrement côté serveur est utilisé AWS Key Management Service comme fournisseur de clés

```
aws emr create-security-configuration --name "MySecConfig" \
    --security-configuration '{
        "EncryptionConfiguration": {
            "EMRWALEncryptionConfiguration":{
                "AwsKmsKey":"arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
                },
            "EnableInTransitEncryption":false, "EnableAtRestEncryption":false
        }
    }'
```

`EnableInTransitEncryption`et pourrait `EnableAtRestEncryption` toujours être vrai, si vous souhaitez activer le chiffrement associé.

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

Paramètre	Description
"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. S'il PEM est spécifié, S3Object il doit s'agir d'une référence à l'emplacement dans Amazon S3 d'un fichier zip contenant les certificats. Si Custom est spécifié, S3Object il doit s'agir d'une référence à l'emplacement d'un fichier JAR dans Amazon S3, suivie d'une CertificateProviderClass entrée.
"S3Object" : " *ZipLocation* " | "*JarLocation* "	Fournit l'emplacement dans Amazon S3 d'un fichier zip lorsqu'il PEM est spécifié, ou d'un fichier JAR lorsque cela Custom est spécifié. Le format peut être un chemin d'accès (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 comporter les fichiers nommés privateKey.pem et certificateChain.pem . Un fichier nommé trustedCertificates.pem est facultatif.
"CertificateProviderClass" : "*MyClassID* "	Obligatoire uniquement si cela Custom est spécifié pourCertificateProviderType . *MyClassID* spécifie un nom de classe complet déclaré dans le fichier JAR, qui implémente l'ArtifactsProvider interface TLS. Par exemple, com.mycompany.MyCertProvider .

Paramètre	Description
Paramètres de chiffrement au repos	
`"AtRestEncryptionConfiguration" :`	Spécifie un ensemble de valeurs pour le chiffrement au repos lorsqu'il `EnableAtRestEncryption` est activé`true`, y compris le chiffrement Amazon S3 et le chiffrement du disque local.
Paramètres de chiffrement Amazon S3	
`"S3EncryptionConfiguration" :`	Spécifie un ensemble de valeurs utilisées pour le chiffrement Amazon S3 avec le système de fichiers Amazon EMR (EMRFS).
`"EncryptionMode" : "SSE-S3" | "SSE-KMS" | "CSE-KMS" | "CSE-Custom"`	Spécifie le type de chiffrement Amazon S3 à utiliser. Si elle `SSE-S3` est spécifiée, aucune autre valeur de chiffrement Amazon S3 n'est requise. Si l'un `SSE-KMS` ou l'autre `CSE-KMS` est spécifié, un AWS KMS key ARN doit être spécifié comme `AwsKmsKey` valeur. 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/12345678-1234-1234-1234-123456789012`).

Paramètre	Description
`"S3Object" : "JarLocation "`	Obligatoire uniquement lorsque `CSE-Custom` cela est spécifié pour`CertificateProviderType` . *`JarLocation`* fournit l'emplacement d'un fichier JAR dans Amazon S3. 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" : "`*`MyS3KeyClassID`*` "`	Obligatoire uniquement lorsque `CSE-Custom` cela est spécifié pour`EncryptionMode` . *`MyS3KeyClassID`* spécifie le nom complet d'une classe déclarée dans l'application qui implémente l' EncryptionMaterialsProviderinterface ; 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.
`"EnableEbsEncryption": true | false`	Spécifiez `true` pour activer le chiffrement EBS. Le chiffrement EBS chiffre le volume du périphérique racine EBS et les volumes de stockage associés. Pour utiliser le chiffrement EBS, vous devez spécifier `AwsKms` comme votre`EncryptionKeyProviderType` .

Paramètre	Description
`"EncryptionKeyProviderType": "AwsKms" | "Custom"`	Spécifie le fournisseur de clés. S'il `AwsKms` est spécifié, un ARN de clé KMS doit être spécifié comme `AwsKmsKey` valeur. Si **`Custom`** est défini, les valeurs `S3Object` et `EncryptionKeyProviderClass` doivent être spécifiées.
`"AwsKmsKey" : "`*`MyKeyARN`*`"`	Obligatoire uniquement lorsque `AwsKms` cela est spécifié pour`Type`. *`MyKeyARN`*doit être un ARN entièrement spécifié pour une clé (par exemple,`arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-456789012123`).
`"S3Object" : "`*`JarLocation`*` "`	Obligatoire uniquement lorsque `CSE-Custom` cela est spécifié pour`CertificateProviderType` . *`JarLocation`* fournit l'emplacement d'un fichier JAR dans Amazon S3. 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 lorsque `Custom` cela est spécifié pour`Type`. *`MyLocalDiskKeyClassID`* spécifie le nom complet d'une classe déclarée dans l'application qui implémente l' EncryptionMaterialsProviderinterface ; par exemple,*`com.mycompany.MyLocalDiskKeyProvider`* .
Paramètres de chiffrement EMR WAL	
`"EMRWALEncryptionConfiguration"`	Spécifie la valeur du chiffrement EMR WAL.

Paramètre	Description
`"AwsKmsKey"`	Spécifie l'identifiant de clé CMK Arn.

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 Utilisation de Kerberos pour l'authentification avec Amazon EMR. 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 paramètre s Kerberos doivent également être spécifiés sur le cluster, sinon une erreur se produira.
Fournisse ur	KDC dédié du cluster	Spécifie qu'Amazon EMR crée un KDC sur le nœud primaire de tout cluster utilisant cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administrateur du KDC lorsque vous créez le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaire. 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 du KDC que vous spécifiez pour le KDC dédié au cluster.

Paramètre		Description
	KDC externe	Disponible uniquement avec les versions Amazon EMR 5.20.0 et supérieures. Spécifie que les clusters utilisant cette configuration de sécurité authentifient les principaux Kerberos à l'aide d'un serveur KDC extérieur au cluster. Aucun KDC n'est créé sur le cluster. Lorsque vous créez le cluster, vous spécifiez le nom de domaine et le mot de passe d'administrateur du KDC pour le KDC externe.
Durée de vie du billet		Facultatif. Spécifie la période pendant 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 primaire pour renouveler un ticket après son expiration.
Relation d'approbation inter-domaines		Spécifie une confiance inter-domaines entre un KDC dédié au cluster sur des clusters utilisant cette configuration de sécurité et un KDC dans un autre domaine Kerberos. Les principaux (généralement les utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est requise. Pour plus d'informations, consultez Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.

Paramètre		Description
Propriétés de confiance entre 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, mais en majuscules.
	Domaine	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.
	Serveur d'administration	Spécifie le FQDN (nom de domaine complet) 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	Spécifie le FQDN (nom de domaine complet) 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		Spécifie que le KDC externe du cluster est utilisé par le cluster.

Paramètre			Description
Propriétés de KDC externe	Serveur d'administration		Spécifie 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		Spécifie 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		Spécifie que l'authentification principale Kerberos est intégrée à un domaine Microsoft Active Directory.
	Propriétés de l'intégration d'Active Directory	Domaine Active Directory	Spécifie le nom de domaine Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine, mais en majuscules.
		Domaine Active Directory	Spécifie le nom du domaine Active Directory.

Paramètre			Description
		Serveur Active Directory	Spécifie le nom de domaine complet (FQDN) du contrôleur de domaine Microsoft Active Directory.

Spécification des paramètres Kerberos à l'aide du 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 Exemples de configuration.

Paramètre			Description
`"AuthenticationConfiguration": {`			Nécessaire pour Kerberos. Spécifie qu'une configuration d'authent ification fait partie de cette configura tion de sécurité.
	`"KerberosConfiguration": {`		Nécessaire pour Kerberos. Spécifie les propriétés de configuration de Kerberos.
		`"Provider": "ClusterD edicatedKdc",` —ou— `"Provider": "ExternalKdc",`	`ClusterDedicatedKdc` indique qu'Amazon EMR crée un KDC sur le nœud primaire de tout cluster utilisant cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administr ateur du KDC lorsque vous créez le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaire. 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 du KDC

Paramètre	Description
	que vous avez spécifiés lorsque vous avez créé le cluster avec le KDC dédié au cluster. *ExternalKdc* indique que le cluster utilise un KDC externe. Amazon EMR ne crée pas de KDC sur le nœud primaire. 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 pendant 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 primaire pour renouveler un ticket après son expiration.

Paramètre	Description
"CrossRealmTrustConfiguration": {	Spécifie une confiance inter-domaines entre un KDC dédié au cluster sur des clusters utilisant cette configuration de sécurité et un KDC dans un autre domaine Kerberos. Les principaux (généralement les utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est requise. Pour plus d'informations, consultez Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.
"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, mais en majuscules.
"Domain": "*kdc2.com*",	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.

Paramètre	Description
"AdminServer": "*kdc.com:749* ",	Spécifie le FQDN (nom de domaine complet) 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* "	Spécifie le FQDN (nom de domaine complet) 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*).
}	
}	

Paramètre	Description
« ExternalKdc Configuration » : {	Obligatoire lorsque *ExternalKdc* est spécifié.
"TicketLifetimeInHours": *24*,	Facultatif. Spécifie la période pendant 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 primaire pour renouveler un ticket après son expiration.
"KdcServerType": "Single",	Spécifie qu'un seul serveur KDC est référencé. `Single` est actuellement la seule valeur prise en charge.

Paramètre	Description
« AdminServer « :" *kdc.com:749* «,	Spécifie 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* «,	Spécifie le nom de domaine complet (FQDN) du serveur KDC externe. Le serveur d'administration et le serveur KDC s'exécutent généralem ent 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*).
`"AdIntegrationConf iguration": {`	Spécifie que l'authentification principale Kerberos est intégrée à un domaine Microsoft Active Directory.

Paramètre	Description
"AdRealm": "*AD.DOMAIN.COM* ",	Spécifie le nom de domaine Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine, mais en majuscules.
"AdDomain": "*ad.domain.com* "	Spécifie le nom du domaine Active Directory.
"AdServer": "*ad.domain.com* "	Spécifie le nom de domaine complet (FQDN) du contrôleur de domaine Microsoft 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 Configuration de rôles IAM pour les demandes EMRFS à Amazon S3.

Spécification des rôles IAM pour EMRFS à l'aide du AWS CLI

Vous trouverez ci-dessous un exemple d'extrait JSON permettant de spécifier des rôles IAM personnalisés pour EMRFS dans une configuration de sécurité. Il montre les mappages de rôles pour les trois types d'identifiants différents, suivis d'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 des mappages de rôles.
`"RoleMappings":`	Obligatoire. Contient une ou plusieurs définitio ns de mappage de rôles. Les mappages de rôles sont évalués dans l'ordre d'apparition du haut vers le bas. Si un mappage de rôle est considéré comme vrai pour un appel de données EMRFS dans Amazon S3, aucun autre mappage de rôle n'est évalué et EMRFS utilise le rôle IAM spécifié pour la demande.

Paramètre	Description
	Les mappages de rôles sont constitués des 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 envoyée à Amazon S3 correspond à l'une des `Identifiers` spécifiées.
`"IdentifierType":`	Les valeurs suivantes sont possibles : • `"User"` indique que les identifiants sont ceux d'un ou de plusieurs utilisateurs Hadoop, qui peuvent être des utilisateurs de comptes Linux ou des utilisateurs principaux de Kerberos. Lorsque la demande EMRFS provient de l'utilisateur ou des utilisateurs spécifiés, le rôle IAM est assumé. • `"Prefix"` indique que l'identifiant est un emplacement Amazon S3. Le rôle IAM est assumé pour les appels vers le ou les emplacements dotés des préfixes spécifiés. Par exemple, le préfixe `s3://mybucket/` correspond à `s3://mybucket/mydir` et `s3://mybucket/yetanotherdir` . • `"Group"` indique que les identifiants sont un ou plusieurs groupes Hadoop. Le rôle IAM est assumé si la demande provient d'un utilisateur appartenant à un ou plusieurs groupes spécifiés.
`"Identifiers":`	Spécifie un ou plusieurs identifiants du type d'identifiant approprié. Séparez les identifiants multiples par des virgules sans espace.

Configuration des demandes de service de métadonnées aux instances Amazon EC2

Les métadonnées d'instance sont des données portant sur votre instance que vous pouvez utiliser pour configurer ou gérer l'instance en cours d'exécution. Vous pouvez accéder aux métadonnées d'instance à partir d'une instance en cours d'exécution en utilisant l'une des méthodes suivantes :

- Service des métadonnées d'instance Version 1 (IMDSv1) – méthode de demande/réponse
- Service des métadonnées d'instance Version 2 (IMDSv2) – méthode orientée session

Alors qu'Amazon EC2 prend en charge à la fois IMDSv1 et IMDSv2, Amazon EMR prend en charge IMDSv2 dans Amazon EMR 5.23.1, 5.27.1, 5.32 ou version ultérieure, et 6.2 ou version ultérieure. Dans ces versions, les composants Amazon EMR utilisent IMDSv2 pour tous les appels IMDS. Pour les appels IMDS dans le code de votre application, vous pouvez utiliser à la fois IMDSv1 et IMDSv2, ou configurer l'IMDS pour utiliser uniquement IMDSv2 pour une sécurité accrue. Lorsque vous spécifiez que IMDSv2 doit être utilisé, IMDSv1 ne fonctionne plus.

Pour plus d'informations, consultez Configurer le service de métadonnées d'instance dans le guide de l'utilisateur Amazon EC2.

> **Note**
>
> Dans les versions antérieures d'Amazon EMR 5.x ou 6.x, la désactivation d'IMDSv1 entraînait l'échec du démarrage du cluster, car les composants Amazon EMR utilisaient IMDSv1 pour tous les appels IMDS. Lorsque vous désactivez IMDSv1, assurez-vous que tout logiciel personnalisé utilisant IMDSv1 est mis à jour vers IMDSv2.

Spécification de la configuration du service de métadonnées d'instance à l'aide de l' AWS CLI

Vous trouverez ci-dessous un exemple d'extrait JSON pour spécifier le service de métadonnées d'instance (IMDS) d'Amazon EC2 dans une configuration de sécurité. L'utilisation d'une configuration de sécurité personnalisée est facultative.

```
{
  "InstanceMetadataServiceConfiguration" : {
      "MinimumInstanceMetadataServiceVersion": integer,
      "HttpPutResponseHopLimit": integer
   }
}
```

Paramètre	Description
`"InstanceMetadataServiceConfiguration":`	Si vous ne spécifiez pas IMDS dans une configuration de sécurité et que vous utilisez une version d'Amazon EMR qui nécessite IMDSv1, Amazon EMR utilise par défaut IMDSv1 comme version minimale du service de métadonnées d'instance. Si vous souhaitez utiliser votre propre configuration, les deux paramètres suivants sont obligatoires.
`"MinimumInstanceMetadataServiceVersion":`	Obligatoire. Spécifiez 1 ou 2. La valeur 1 autorise IMDSv1 et IMDSv2. La valeur 2 autorise uniquement IMDSv2.
`"HttpPutResponseHopLimit":`	Obligatoire. Limite de saut de réponse HTTP PUT souhaitée pour les requêtes de métadonnées d'instance. Plus le nombre est élevé, plus les demandes de métadonnées d'instance peuvent être envoyées. Par défaut: 1. Spécifiez un nombre entier compris entre 1 et 64.

Spécification de la configuration du service de métadonnées d'instance à l'aide de la console

Vous pouvez configurer l'utilisation d'IMDS pour un cluster lorsque vous le lancez depuis la console Amazon EMR.
Contrôles des configurations de sécurité IMDS dans la console Amazon EMR

Pour configurer l'utilisation d'IMDS à l'aide de la console :

1. Lorsque vous créez une nouvelle configuration de sécurité sur la page Configurations de sécurité, sélectionnez Configurer le service de métadonnées d'instance EC2 sous le paramètre Service de métadonnées d'instance EC2. Cette configuration est prise en charge uniquement dans Amazon EMR 5.23.1, 5.27.1, 5.32 ou version ultérieure, et 6.2 ou version ultérieure.
2. Pour l'option Version minimale du service de métadonnées d'instance, sélectionnez l'une des options suivantes :

- Désactiver IMDSv1 et autoriser uniquement IMDSv2, si vous voulez autoriser uniquement IMDSv2 sur ce cluster. Consultez la section Transition vers l'utilisation du service de métadonnées d'instance version 2 dans le guide de l'utilisateur Amazon EC2.
- Autoriser IMDSv1 et IMDSv2 sur le cluster, si vous voulez autoriser IMDSv1 et IMDSv2 orienté session sur ce cluster.

3. Pour IMDSv2, vous pouvez également configurer le nombre de sauts de réseau à réseau autorisés pour le jeton de métadonnées en définissant la limite de sauts de réponse put HTTP sur un entier compris entre 1 et 64.

Pour plus d'informations, consultez Configurer le service de métadonnées d'instance dans le guide de l'utilisateur Amazon EC2.

Consultez Configurer les détails de l'instance et Configurer le service de métadonnées de l'instance dans le guide de l'utilisateur Amazon EC2.

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. Vous pouvez utiliser le AWS Management Console ou le AWS CLI.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour définir une configuration de sécurité à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Configuration de sécurité et autorisations, recherchez le champ Configuration de sécurité. Sélectionnez le menu déroulant ou choisissez Parcourir pour sélectionner le nom

d'une configuration de sécurité que vous avez créée précédemment. Vous pouvez également choisir Créer une configuration de sécurité pour créer une configuration que vous pouvez utiliser pour votre cluster.

4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour définir une configuration de sécurité à l'aide de l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Créer un cluster et Go to advanced options (Aller aux options avancées).
3. Sur l'écran Étape 1 : Logiciel et étapes, dans la liste Version, sélectionnez emr-4.8.0 ou une version plus récente. Choisissez les paramètres que vous voulez, puis cliquez sur Next (Suivant).
4. Sur l'écran Step 2: Hardware (Étape 2 : Matériel), sélectionnez les paramètres que vous voulez, puis cliquez sur Next (Suivant). Faites la même chose pour Step 3: General Cluster Settings (Étape 3 : Paramètres généraux de cluster).
5. Sur l'écran Step 4: Security (Étape 4 : Sécurité), sous Encryption Options (Options de chiffrement), choisissez une valeur pour Security configuration (Configuration de sécurité).
6. Configurez les autres options de sécurité comme vous le souhaitez et choisissez Create cluster (Créer le cluster).

CLI

Pour définir une configuration de sécurité à l'aide du AWS CLI

- Utilisez `aws emr create-cluster` pour 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 version `--release-label` que vous spécifiez 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

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans Amazon EMR. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure mondiale qui gère l'ensemble du AWS cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu inclut la configuration de la sécurité et les tâches de gestion pour le AWS produit que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez les FAQ sur la confidentialité des données. Pour plus d'informations sur la protection des données en Europe, consultez le modèle de responsabilité partagée d'Amazon et le billet de blog sur le RGPD sur le blog sur la AWS sécurité.

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

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole TLS pour communiquer avec les AWS ressources. Nous avons besoin du protocole TLS 1.2.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut au sein AWS des services.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une CLI ou une API, utilisez un point de terminaison FIPS. Pour de plus amples informations sur les points de terminaison FIPS disponibles, consultez Federal Information Processing Standard (FIPS) 140-2.

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 inclut lorsque vous travaillez avec Amazon EMR ou d'autres AWS services à l'aide de la console, de l'API ou AWS des AWS CLI SDK. Toutes les données que vous saisissez dans Amazon EMR ou d'autres services peuvent être récupérées pour 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.

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 des données pour les clusters. Les configurations de sécurité proposent des paramètres pour activer la sécurité des données en transit et au repos dans les volumes Amazon Elastic Block Store (Amazon EBS) et d'EMRFS dans 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 Guide de mise à jour Amazon EMR.

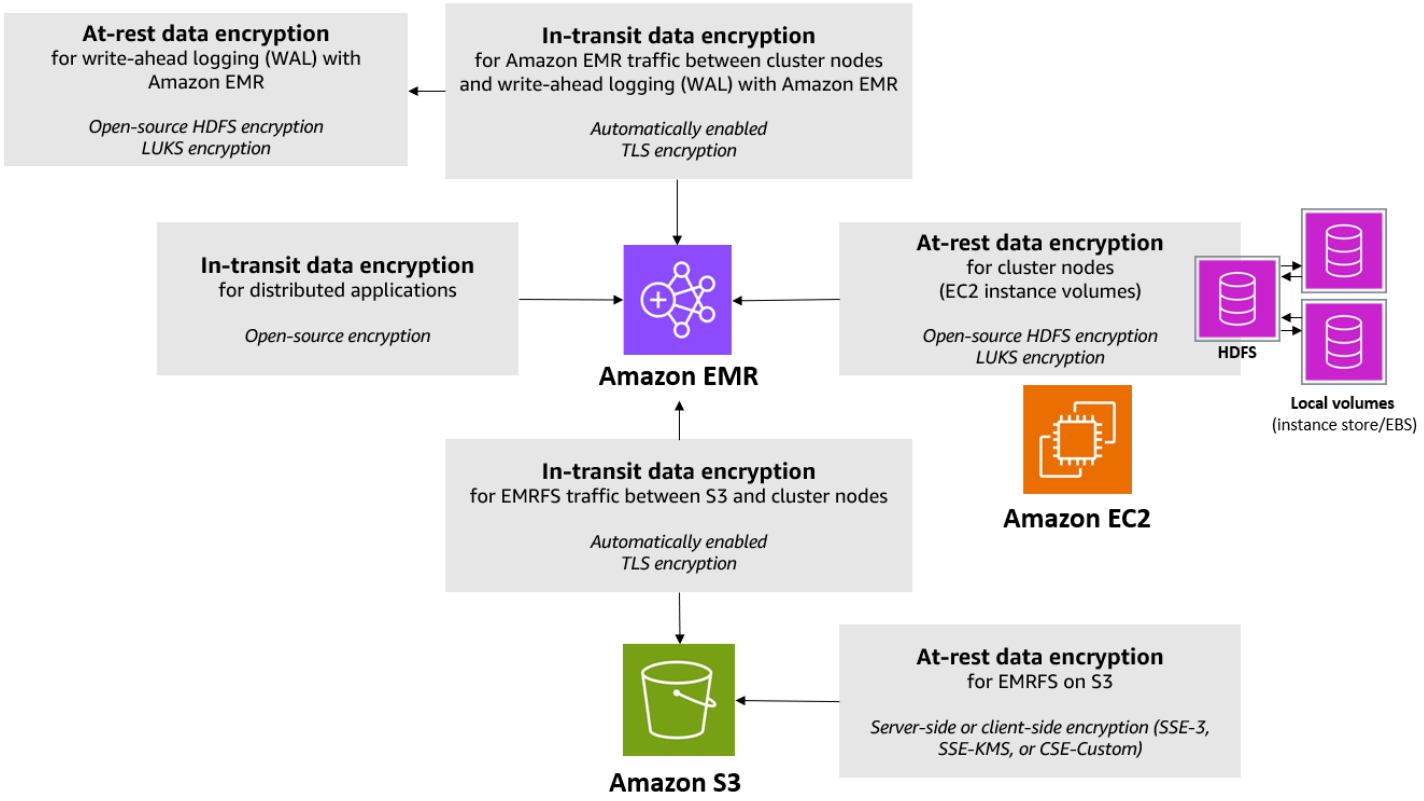
Rubriques

- Options de chiffrement
- Création de clés et de certificats pour le chiffrement des données

Options de chiffrement

Avec les versions 4.8.0 et supérieures 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 les 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 des données chaque fois qu'un cluster est créé. Pour plus d'informations, consultez Création d'une configuration de sécurité.

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 Guide de mise à jour Amazon EMR.
- 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 à l'aide des propriétés EMRFS.
- 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 Guide de gestion Amazon EMR.

> **Note**
>
> À partir de la version 5.24.0 d'Amazon EMR, vous pouvez utiliser une option de configuration de sécurité pour chiffrer le périphérique racine EBS et les volumes de stockage lorsque vous le spécifiez comme fournisseur de clés. AWS KMS Pour plus d'informations, consultez Chiffrement de disque local.

Le chiffrement des données nécessite des clés et des certificats. Une configuration de sécurité vous donne la flexibilité de choisir entre plusieurs options, notamment les clés gérées par AWS Key Management Service, les clés gérées par Amazon S3 et les clés et certificats fournis par les fournisseurs personnalisés que vous fournissez. Lorsque vous l'utilisez en AWS KMS tant que fournisseur de clés, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour en savoir plus, consultez AWS KMS Tarification.

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 des données EMRFS dans Amazon S3

Le chiffrement Amazon S3 fonctionne avec les objets du système de fichiers Amazon EMR (EMRFS) lus et écrits sur Amazon S3. Vous indiquez le chiffrement côté serveur (SSE) ou le chiffrement côté client (CSE) sur 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 plus d'informations sur le chiffrement Amazon S3, consultez la section Protection des données à l'aide du chiffrement dans le guide de l'utilisateur d'Amazon Simple Storage Service.

> **Note**
>
> Lorsque vous les utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour plus d'informations, consultez Tarification d'AWS KMS.

Chiffrement côté serveur sur Amazon S3

Lorsque vous configurez le chiffrement côté serveur sur Amazon S3, Amazon S3 chiffre les données au niveau de l'objet au moment où elles sont écrites sur le disque et déchiffre les données lorsqu'elles sont accédées. Pour plus d'informations sur SSE, consultez Protection des données à l'aide du chiffrement côté serveur dans le Guide de l'utilisateur Amazon Simple Storage Service.

Lorsque vous indiquez le chiffrement SSE sur Amazon EMR, vous pouvez choisir entre deux systèmes de gestion de clés différents :

- SSE-S3 – Amazon S3 gère les clés pour vous.
- SSE-KMS — Vous utilisez un AWS KMS key pour configurer des politiques adaptées à Amazon EMR. Pour plus d'informations sur les exigences clés relatives à Amazon EMR, consultez la section Utilisation à des AWS KMS keys fins de chiffrement.

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

Chiffrement côté client sur Amazon S3

Avec le chiffrement côté client sur Amazon S3, le chiffrement et le déchiffrement par Amazon S3 se déroulent dans le client EMRFS de votre cluster. Les objets sont chiffrés avant d'être chargés sur Amazon S3 et déchiffrés après leur chargement. Le fournisseur que vous indiquez fournit la clé de chiffrement utilisée par le client. Le client peut utiliser les clés fournies par AWS KMS (CSE-KMS) ou une classe Java personnalisée qui fournit la clé racine côté client (CSE-C). Les spécificités du chiffrement sont légèrement différentes entre CSE-KMS et CSE-C, en fonction du fournisseur indiqué et des métadonnées de l'objet à déchiffrer ou à chiffrer. Pour plus d'informations sur ces différences, consultez Protection des données à l'aide du chiffrement côté client dans le Guide de l'utilisateur Amazon Simple Storage Service.

> **Note**
>
> Le chiffrement CSE sur Amazon S3 garantit uniquement que les données EMRFS échangées avec Amazon S3 sont chiffrées ; cela ne signifie pas que toutes les données sur les volumes des instances du cluster sont chiffrées. De plus, étant donné que Hue n'utilise pas EMRFS, les objets que le navigateur de fichiers S3 de Hue écrit sur Amazon S3 ne sont pas chiffrés.

Chiffrement au repos pour les données dans Amazon EMR WAL

Lorsque vous configurez le chiffrement côté serveur (SSE) pour la journalisation par écriture anticipée (WAL), Amazon EMR chiffre les données au repos. Lorsque vous spécifiez SSE dans Amazon EMR, vous pouvez choisir entre deux systèmes de gestion des clés différents :

SSE-EMR-WAL

Amazon EMR gère les clés pour vous. Par défaut, Amazon EMR chiffre les données que vous avez stockées dans Amazon EMR WAL. SSE-EMR-WAL

SSE-KMS-WAL

Vous utilisez une AWS KMS clé pour configurer les politiques qui s'appliquent à Amazon EMR WAL. Pour plus d'informations sur les principales exigences relatives à Amazon EMR, consultez. Utilisation AWS KMS keys pour le chiffrement

Vous ne pouvez pas utiliser votre propre clé avec SSE lorsque vous activez le WAL avec Amazon EMR. Pour plus d'informations, consultez la section Write ahead logs (WAL) pour Amazon EMR.

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 activant le chiffrement en transit. Pour plus d'informations, consultez Chiffrement en transit. 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 Guide de mise à jour Amazon EMR.

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 la section sur les volumes SSD NVMe dans le guide de l'utilisateur Amazon EC2. 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 la section Chiffrement par défaut dans le guide de l'utilisateur Amazon EC2. Lorsque le chiffrement du disque local est activé dans une configuration de sécurité, les paramètres Amazon EMR ont priorité sur les paramètres Amazon EC2 pour les instances du cluster EC2 encryption-by-default .

Les options suivantes sont disponibles pour chiffrer les volumes EBS à l'aide d'une configuration de sécurité :

- Chiffrement EBS : à partir d'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 n'est disponible que lorsque vous le 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 votre fournisseur de clés, vous pouvez configurer une AWS KMS key avec des politiques adaptées à Amazon EMR, ou une classe Java personnalisée qui fournit les artefacts de chiffrement. Lorsque vous les utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour en savoir plus, consultez AWS KMS Tarification.

> **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 d'application Apache. Pour plus d'informations, consultez Configuration des applications.

Hadoop

- Le shuffle MapReduce chiffré Hadoop 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).
- Pour plus d'informations, consultez Hadoop en mode sécurisé dans la documentation Apache Hadoop.

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 Utilisation de Kerberos pour l'authentification avec Amazon EMR.

Hive

- Les communications du client JDBC/ODBC avec HiveServer 2 (HS2) sont chiffrées à l'aide des configurations SSL dans les versions 6.9.0 et ultérieures d'Amazon EMR.
- Pour plus d'informations, consultez la section relative au chiffrement SSL de la documentation Apache Hive.

Spark

- La communication RPC interne entre les composants Spark, tels que le service de transfert de blocs et le service de mélange externe, est chiffrée à l'aide du code AES-256 dans les versions 5.9.0 et ultérieures d'Amazon EMR. 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.
- Pour plus d'informations, consultez la section Paramètres de sécurité Spark de la documentation Apache Spark.

Tez

- Gestionnaire de mélange de Tez utilise le protocole TLS (`tez.runtime.ssl.enable`).

Presto

- Les communications internes entre les nœuds Presto utilisent SSL/TLS (Amazon EMR version 5.6.0 et ultérieures uniquement).

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.

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 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 les utilisez AWS KMS, des frais s'appliquent pour le stockage et l'utilisation des clés de chiffrement. Pour en savoir plus, consultez AWS KMS Tarification.

Cette rubrique fournit des informations sur la stratégie de clé 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 Guide du développeur AWS Key Management Service .

Utilisation AWS KMS keys pour le chiffrement

La clé de AWS KMS chiffrement 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 se trouve dans un compte différent de celui que vous utilisez pour configurer un cluster, vous devez spécifier la clé à l'aide de son ARN.

Le rôle du profil d'instance Amazon EC2 doit être autorisé à utiliser la clé KMS que vous spécifiez. Le rôle par défaut du profil d'instance dans Amazon EMR est `EMR_EC2_DefaultRole`. Si vous utilisez un rôle différent pour le profil d'instance, ou si vous utilisez des rôles IAM pour les demandes EMRFS adressées à Amazon S3, assurez-vous que chaque rôle est ajouté en tant qu'utilisateur clé, le cas échéant. Cela donne au rôle des autorisations pour utiliser la clé KMS. Pour plus d'informations, consultez les sections Utilisation des stratégies de clé dans le Guide du développeur AWS Key Management Service et Configuration des rôles IAM pour les demandes EMRFS adressées à Amazon S3.

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

Notez qu'Amazon EMR prend uniquement en charge les clés KMS symétriques. Vous ne pouvez pas utiliser une clé KMS asymétrique pour chiffrer les données au repos dans un cluster Amazon EMR. Pour savoir si une clés KMS est symétrique ou asymétrique, consultez Identification de clés KMS symétriques et asymétriques.

La procédure ci-dessous décrit comment ajouter le profil d'instance Amazon EMR par défaut, `EMR_EC2_DefaultRole`, en tant qu'utilisateur de clé à l'aide de la AWS Management Console. Elle suppose que vous avez déjà créé une clé KMS. Pour créer une nouvelle clé KMS, consultez Création de clés dans le Guide du développeur AWS Key Management Service .

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

1. Connectez-vous à la console AWS Key Management Service (AWS KMS) AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Sélectionnez l'alias de la clé KMS 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 KMS

À partir de la version 5.24.0 d'Amazon EMR, 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 AWS KMS le spécifier comme fournisseur principal. En outre, vous devez accorder au rôle de service `EMR_DefaultRole` les autorisations nécessaires pour utiliser celles AWS KMS key que vous spécifiez.

Vous pouvez utiliser le AWS Management Console pour ajouter le rôle de service à la liste des utilisateurs clés pour la clé KMS spécifiée, ou vous pouvez utiliser le AWS CLI ou un AWS SDK pour associer une politique de clé appropriée.

La procédure suivante décrit comment utiliser le AWS Management Console pour ajouter le rôle de service Amazon EMR par défaut en `EMR_DefaultRole` tant qu'utilisateur clé. Elle suppose que vous avez déjà créé une clé KMS. Pour créer une nouvelle clé KMS, consultez Création de clés dans le Guide du développeur AWS Key Management Service .

Pour ajouter le rôle de service Amazon EMR à la liste des utilisateurs de clés de chiffrement

1. Connectez-vous à la console AWS Key Management Service (AWS KMS) AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Sélectionnez Clés gérées par le client dans la barre latérale gauche.
4. Sélectionnez l'alias de la clé KMS à modifier.
5. Sur la page de détails de la clé, sous Key Users (Utilisateurs de clés), choisissez Add (Ajouter).
6. Dans la section Ajouter des utilisateurs clés, sélectionnez le rôle approprié. Le nom du rôle de service par défaut pour Amazon EMR est. `EMR_DefaultRole`
7. Choisissez 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. Les exigences relatives au fournisseur de clés personnalisées dépendent de l'utilisation du chiffrement de disque local et du chiffrement Amazon S3, ainsi que de la version publiée par Amazon EMR.

Selon le type de chiffrement que vous utilisez lors de la création d'un fournisseur de clés personnalisé, l'application doit également implémenter différentes EncryptionMaterialsProvider interfaces. Les deux interfaces sont disponibles dans le AWS SDK pour Java version 1.11.0 et versions ultérieures.

- Pour implémenter le chiffrement Amazon S3, utilisez le fichier com.amazonaws.services.s3.model. EncryptionMaterialsProvider interface.
- Pour implémenter le chiffrement du disque local, utilisez le fichier com.amazonaws.services.elasticmapreduce.spi.security. EncryptionMaterialsProvider interface.

Vous pouvez utiliser n'importe quelle stratégie pour fournir du matériel de chiffrement pour la mise en œuvre. Par exemple, vous pouvez choisir de fournir du matériel de chiffrement statique ou de l'intégrer à un système de gestion de clés plus complexe.

Si vous utilisez le chiffrement Amazon S3, vous devez utiliser les algorithmes de chiffrement AES/GCM/ NoPadding pour le matériel de chiffrement personnalisé.

Si vous utilisez le chiffrement de disque local, l'algorithme de chiffrement à utiliser pour le matériel de chiffrement personnalisé varie selon la version de l'EMR. Pour Amazon EMR 7.0.0 et versions antérieures, vous devez utiliser AES/GCM/. NoPadding Pour Amazon EMR 7.1.0 et versions ultérieures, vous devez utiliser AES.

La EncryptionMaterialsProvider classe obtient le matériel de chiffrement par contexte de chiffrement. Amazon EMR renseigne les informations contextuelles de chiffrement au moment de l'exécution pour aider l'appelant à déterminer les matériaux de chiffrement à renvoyer.

Example Exemple : utilisation d'un fournisseur de clés personnalisé pour le chiffrement Amazon S3 avec EMRFS

Lorsqu'Amazon EMR extrait les matériaux de chiffrement de la EncryptionMaterialsProvider classe pour effectuer le chiffrement, EMRFS remplit éventuellement l'argument MaterialsDescription avec deux champs : l'URI Amazon S3 de l'objet et JobFlowId l'URI du cluster, qui peuvent être utilisés par la classe pour renvoyer du matériel de chiffrement de manière sélective. EncryptionMaterialsProvider

Par exemple, le fournisseur peut renvoyer des clés différentes pour différents préfixes d'URI Amazon S3. C'est la description des matériaux de chiffrement renvoyés qui 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 du matériel de chiffrement est transmise à la EncryptionMaterialsProvider classe, afin qu'elle puisse, à nouveau, renvoyer de manière sélective la clé correspondante pour déchiffrer l'objet.

Une implémentation EncryptionMaterialsProvider de référence est fournie ci-dessous. Un autre fournisseur personnalisé, le fournisseur EMRFSRSA, est EncryptionMaterials disponible auprès 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 4.8.0 d'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. La classe doit implémenter l'ArtifactsProviderinterface TLS disponible à partir de la AWS SDK for Java version 1.11.0.

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.

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	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

Nom de fichier	Obligatoire/facultatif	Détails
		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 cluster se trouve dans la région us-west-2 (Oregon), vous pouvez spécifier `CN=*.us-west-2.compute.internal`.

Si le fichier PEM fourni dans l'artefact de chiffrement ne contient pas de caractère générique dans le CN du domaine, vous devez modifier la valeur de `hadoop.ssl.hostname.verifier` en `ALLOW_ALL`. Cela se fait avec la classification `core-site` lors de la soumission de configurations à un cluster ou en ajoutant cette valeur dans le fichier `core-site.xml`. Cette modification est nécessaire car le vérificateur de nom d'hôte par défaut n'accepte pas un nom d'hôte sans le caractère générique, ce qui entraîne une erreur. Pour plus d'informations sur la configuration d'un cluster EMR au sein d'un Amazon VPC, consultez Configuration de la mise en réseau.

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 n'est qu'une proof-of-concept démonstration. 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 outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs IAM contrôlent les personnes qui peuvent être authentifiées (connectées) et autorisées (disposant d'autorisations) à utiliser des ressources Amazon EMR. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Fonctionnement d'Amazon EMR avec IAM
- Rôles d'exécution pour les étapes Amazon EMR
- Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS .
- Exemples de politiques basées sur une identité pour Amazon EMR

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Amazon EMR.

Utilisateur du service : si vous utilisez le service Amazon EMR service pour accomplir votre tâche, votre administrateur vous fournira les informations d'identification et les autorisations nécessaires. Vous pourrez avoir besoin d'autorisations supplémentaires si vous utilisez davantage de fonctions

Amazon EMR. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans Amazon EMR, consultez Résolution de problèmes pour identité et accès Amazon EMR.

Administrateur du service – Si vous êtes le responsable des ressources Amazon EMR de votre entreprise, vous bénéficiez probablement d'un accès total à Amazon EMR. C'est à vous de déterminer les fonctions et les ressources Amazon EMR auxquelles vos utilisateurs des services 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 découvrir la façon dont votre entreprise peut utiliser IAM avec Amazon EMR, consultez Fonctionnement d'Amazon EMR avec IAM.

Administrateur IAM : si vous êtes un administrateur IAM, vous souhaiterez peut-être obtenir des informations sur la façon dont vous pouvez écrire des politiques pour gérer l'accès à Amazon EMR. Pour afficher des exemples de politiques basées sur l'identité Amazon EMR que vous pouvez utiliser dans IAM, consultez Exemples de politiques basées sur une identité pour Amazon EMR (Exemples de politiques basées sur l'identité pour Amazon Simple Notification Service).

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide

de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre

source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne 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 IAMAdmins et accorder à ce groupe les autorisations 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 en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie,

l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez 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 Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques 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 politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d’accès (ACL)

Les listes de contrôle d’accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l’autorisation d’accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d’accès, consultez Vue d’ensemble des listes de contrôle d’accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d’autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d’autorisations : une limite d’autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d’autorisations qu’une politique 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 en résultant représentent la combinaison des politiques basées sur l’identité d’une entité et de ses limites d’autorisation. Les politiques 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 politiques remplace l’autorisation. Pour plus d’informations sur les limites d’autorisations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.
- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d’une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l’un ou à l’ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d’informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l’utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques

basées sur l’identité de l’utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d’une politique basée sur les ressources. Un refus explicite dans l’une de ces politiques annule l’autorisation. Pour plus d’informations, consultez politiques de séance dans le Guide de l’utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s’appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Fonctionnement d'Amazon EMR avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon EMR, découvrez les fonctionnalités IAM qui peuvent être utilisées avec Amazon EMR.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon EMR

Fonction IAM	Support d'Amazon EMR
Politiques basées sur l’identité	Oui
Politiques basées sur les ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition d’une politique	Oui
ACL	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Fonctions du service	Non

Fonction IAM	Support d'Amazon EMR
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont Amazon EMR et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon EMR

Prend en charge les politiques basées sur l’identité	Oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Création de politiques IAM dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Exemples de politiques basées sur une identité pour Amazon EMR

Pour voir des exemples de politiques Amazon EMR basées sur l'identité, consultez Exemples de politiques basées sur une identité pour Amazon EMR.

Politiques basées sur les ressources au sein d'Amazon EMR

Prend en charge les politiques basées sur les ressources	Oui

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions de politique pour Amazon EMR

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d’actions dans une stratégie afin d’accorder l’autorisation d’exécuter les opérations associées.

Pour afficher la liste des actions Amazon EMR, consultez Actions, ressources et clés de condition pour Amazon EMR dans la Référence de l'autorisation de service.

Les actions de politique dans Amazon EMR utilisent le préfixe suivant avant l'action :

```
EMR
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "EMR:action1",
      "EMR:action2"
         ]
```

Pour voir des exemples de politiques Amazon EMR basées sur l'identité, consultez Exemples de politiques basées sur une identité pour Amazon EMR.

Ressources de politique pour Amazon EMR

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C’est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L’élément de politique JSON `Resource` indique le ou les objets auxquels l’action s’applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l’aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d’indiquer que l’instruction s’applique à toutes les ressources.

```
"Resource": "*"
```

Pour afficher la liste des types de ressources Amazon EMR et de leurs ARN, consultez Ressources définies par Amazon EMR dans la Référence de l'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier pour l'ARN de chaque ressource, consultez Actions, ressources et clés de condition pour Amazon EMR.

Pour voir des exemples de politiques Amazon EMR basées sur l'identité, consultez Exemples de politiques basées sur une identité pour Amazon EMR.

Clés de condition de politique pour Amazon EMR

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

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, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique 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 `OR` opération 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 Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition d'Amazon EMR et savoir quelles actions et ressources vous pouvez utiliser avec une clé de condition, consultez Actions, ressources et clés de condition pour Amazon EMR dans la Référence de l'autorisation de service.

Pour voir des exemples de politiques Amazon EMR basées sur l'identité, consultez Exemples de politiques basées sur une identité pour Amazon EMR.

Listes de contrôle d'accès (ACL) dans Amazon EMR

Prend en charge les listes ACL	Non

Les listes de contrôle d’accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l’autorisation d’accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

Contrôle d'accès par attributs (ABAC) avec Amazon EMR

Prend en charge ABAC (étiquettes dans les politiques)	Oui

Le contrôle d’accès par attributs (ABAC) est une stratégie d’autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L’étiquetage des entités et des ressources est la première étape d’ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l’identification du principal correspond à celle de la ressource à laquelle il tente d’accéder.

L’ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec Amazon EMR

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations de principal entre services pour Amazon EMR

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d’autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d’accès.

Fonctions du service pour Amazon EMR

Prend en charge les fonctions de service	Non

Rôles liés à un service pour Amazon EMR

Prend en charge les rôles liés à un service.	Oui

Pour plus d’informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Utiliser les balises de cluster et de bloc-notes avec des politiques IAM de contrôle d'accès

Les autorisations pour les actions Amazon EMR associées aux blocs-notes EMR et aux clusters EMR peuvent être ajustées à l'aide d'un contrôle d'accès basé sur des balises avec les politiques 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 ModifyInstanceGroupes ne prend pas en charge les clés contextuelles car il `ClusterID` s'agit d'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 politiques basées sur une identité pour Amazon EMR.
- 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.

Exemples

Pour afficher la liste des actions Amazon EMR, consultez Actions définies par Amazon EMR dans le Guide de l'utilisateur IAM.

Rôles d'exécution pour les étapes Amazon EMR

Un rôle d'exécution est un rôle AWS Identity and Access Management (IAM) que vous pouvez spécifier lorsque vous soumettez une tâche ou une requête à un cluster Amazon EMR. La tâche ou

la requête que vous soumettez à votre cluster Amazon EMR utilise le rôle d'exécution pour accéder à AWS des ressources, telles que des objets dans Amazon S3. Vous pouvez spécifier des rôles d'exécution avec Amazon EMR pour les tâches Spark et Hive.

Vous pouvez également spécifier des rôles d'exécution lorsque vous vous connectez à des clusters Amazon EMR dans Amazon SageMaker et lorsque vous attachez un espace de travail Amazon EMR Studio à un cluster EMR. Pour plus d'informations, consultez Se connecter à un cluster Amazon EMR depuis Studio et et Exécuter un Workspace EMR Studio avec un rôle d'exécution.

Auparavant, les clusters Amazon EMR exécutaient des tâches ou des requêtes Amazon EMR avec des autorisations basées sur la politique IAM attachée au profil d'instance que vous utilisiez pour lancer le cluster. Cela signifiait que les politiques devaient contenir l'union de toutes les autorisations pour toutes les tâches et requêtes exécutées sur un cluster Amazon EMR. Avec les rôles d'exécution, vous pouvez désormais gérer le contrôle d'accès pour chaque tâche ou requête individuellement, au lieu de partager le profil d'instance Amazon EMR du cluster.

Sur les clusters Amazon EMR dotés de rôles d'exécution, vous pouvez également appliquer un contrôle d'accès AWS Lake Formation basé aux tâches et requêtes Spark, Hive et Presto concernant vos lacs de données. Pour en savoir plus sur la façon de procéder à l'intégration à AWS Lake Formation, voirIntégrez Amazon EMR à AWS Lake Formation.

> **Note**
>
> Lorsque vous spécifiez un rôle d'exécution pour une étape Amazon EMR, les tâches ou requêtes que vous soumettez ne peuvent accéder qu'aux AWS ressources autorisées par les politiques associées au rôle d'exécution. Ces tâches et requêtes ne peuvent pas accéder au service de métadonnées d'instance sur les instances EC2 du cluster ni utiliser le profil d'instance EC2 du cluster pour accéder aux ressources AWS .

Conditions préalables au lancement d'un cluster Amazon EMR doté d'un rôle d'exécution

Rubriques

- Étape 1 : Configurer la sécurité dans Amazon EMR
- Étape 2 : Configurer un profil d'instance EC2 pour le cluster Amazon EMR
- Étape 3 : Configurer une politique d'approbation

Étape 1 : Configurer la sécurité dans Amazon EMR

Utilisez la structure JSON suivante pour créer une configuration de sécurité sur le AWS Command Line Interface (AWS CLI), et définissez `EnableApplicationScopedIAMRole` sur`true`. Pour plus d'informations sur les configurations de sécurité, consultez Utilisation de configurations de sécurité pour configurer la sécurité du cluster.

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true
        }
    }
}
```

Nous vous recommandons de toujours activer les options de chiffrement en transit dans la configuration de sécurité, afin que les données transférées sur Internet soient chiffrées, plutôt qu'en texte brut. Vous pouvez ignorer ces options si vous ne souhaitez pas vous connecter aux clusters Amazon EMR avec des rôles d'exécution issus de SageMaker Runtime Studio ou EMR Studio. Pour configurer le chiffrement des données, consultez Configuration du chiffrement des données.

Vous pouvez également créer une configuration de sécurité avec des paramètres personnalisés à l'aide de la AWS Management Console.

Étape 2 : Configurer un profil d'instance EC2 pour le cluster Amazon EMR

Les clusters Amazon EMR utilisent le rôle de profil d'instance Amazon EC2 pour assumer les rôles d'exécution. Pour utiliser des rôles d'exécution avec les étapes Amazon EMR, ajoutez les politiques suivantes au rôle IAM que vous prévoyez d'utiliser comme rôle de profil d'instance. Pour ajouter des politiques à un rôle IAM ou modifier une politique en ligne ou gérée existante, consultez Ajout et suppression d'autorisations d'identité IAM.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowRuntimeRoleUsage",
            "Effect":"Allow",
            "Action":[
                "sts:AssumeRole",
                "sts:TagSession"
```

```
            ],
            "Resource":[
                <runtime-role-ARN>
            ]
        }
    ]
}
```

Étape 3 : Configurer une politique d'approbation

Pour chaque rôle IAM que vous prévoyez d'utiliser comme rôle d'exécution, définissez la politique d'approbation suivante, en remplaçant `EMR_EC2_DefaultRole` par votre rôle de profil d'instance. Pour modifier la politique d'approbation d'un rôle IAM, consultez Modification d'une politique d'approbation de rôle.

```
{
    "Sid":"AllowAssumeRole",
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":"sts:AssumeRole"
}
```

Lancement d'un cluster Amazon EMR avec un contrôle d'accès basé sur les rôles

Après avoir mis en place vos configurations, vous pouvez lancer un cluster Amazon EMR avec la configuration de sécurité depuis Étape 1 : Configurer la sécurité dans Amazon EMR. Pour utiliser des rôles d'exécution avec les étapes Amazon EMR, utilisez l'étiquette de version `emr-6.7.0` ou une version ultérieure, et sélectionnez Hive, Spark ou les deux comme application de cluster. Pour vous connecter depuis SageMaker Studio, utilisez release `emr-6.9.0` ou version ultérieure et sélectionnez Livy, Spark, Hive ou Presto comme application de cluster. Pour savoir comment mettre à jour votre cluster, consultez Spécification d'une configuration de sécurité pour un cluster.

Soumission de tâches Spark à l'aide des étapes d'Amazon EMR

Voici un exemple d'exécution de l' HdfsTest exemple inclus dans Apache Spark. Cet appel d'API ne réussit que si le rôle d'exécution Amazon EMR fourni peut accéder à `S3_LOCATION`.

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
```

```
S3_LOCATION=<s3-path>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Spark Example", "ActionOnFailure": "CONTINUE","HadoopJarStep":
 { "Jar":"command-runner.jar","Args" : ["spark-example","HdfsTest",
 "$S3_LOCATION"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

> **Note**
>
> Nous vous recommandons de désactiver l'accès SSH au cluster Amazon EMR et d'autoriser uniquement l'API Amazon EMR `AddJobFlowSteps` à accéder au cluster.

Soumission de tâches Hive à l'aide des étapes d'Amazon EMR

L'exemple suivant utilise Apache Hive avec les étapes Amazon EMR pour soumettre une tâche afin d'exécuter le fichier `QUERY_FILE.hql`. Cette requête n'aboutit que si le rôle d'exécution fourni peut accéder au chemin Amazon S3 du fichier de requête.

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Run hive query using command-runner.jar - simple
 select","ActionOnFailure":"CONTINUE","HadoopJarStep": { "Jar": "command-
runner.jar","Args" :["hive -
f","s3://DOC_EXAMPLE_BUCKET/QUERY_FILE.hql"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

Connectez-vous aux clusters Amazon EMR avec des rôles d'exécution depuis un SageMaker bloc-notes Studio

Vous pouvez appliquer des rôles d'exécution Amazon EMR aux requêtes que vous exécutez dans des clusters Amazon EMR depuis Studio. SageMaker Pour ce faire, suivez les étapes suivantes.

1. Suivez les instructions de la section Lancer Amazon SageMaker Studio pour créer un SageMaker studio.
2. Dans l'interface utilisateur de SageMaker Studio, démarrez un bloc-notes avec des noyaux compatibles. Par exemple, démarrez une SparkMagic image avec un PySpark noyau.
3. Choisissez un cluster Amazon EMR dans SageMaker Studio, puis sélectionnez Connect.
4. Choisissez un rôle d'exécution, puis sélectionnez Connecter.

Cela créera une cellule de SageMaker bloc-notes avec des commandes magiques pour se connecter à votre cluster Amazon EMR avec le rôle d'exécution Amazon EMR choisi. Dans la cellule du bloc-notes, vous pouvez saisir et exécuter des requêtes avec un rôle d'exécution et un contrôle d'accès basé sur Lake Formation. Pour un exemple plus détaillé, consultez Appliquer des contrôles d'accès aux données précis avec AWS Lake Formation Amazon EMR depuis Amazon Studio. SageMaker

Contrôle de l'accès au rôle d'exécution Amazon EMR

Vous pouvez contrôler l'accès au rôle d'exécution à l'aide de la clé de condition `elasticmapreduce:ExecutionRoleArn`. La politique suivante permet à un principal IAM d'utiliser un rôle IAM nommé `Caller`, ou tout autre rôle IAM commençant par la chaîne `CallerTeamRole`, comme rôle d'exécution.

> **Important**
>
> Vous devez créer une condition basée sur la clé de contexte `elasticmapreduce:ExecutionRoleArn` lorsque vous autorisez un appelant à appeler les API `AddJobFlowSteps` ou `GetClusterSessionCredentials`, comme le montre l'exemple suivant.

```
{
    "Sid":"AddStepsWithSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:AddJobFlowSteps"
    ],
    "Resource":"*",
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
```

```
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/Caller"
            ]
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/CallerTeamRole*"
            ]
        }
    }
}
```

Établir la confiance entre les rôles d'exécution et les clusters Amazon EMR

Amazon EMR génère un identifiant unique `ExternalId` pour chaque configuration de sécurité avec une autorisation de rôle d'exécution activée. Cette autorisation permet à chaque utilisateur de posséder un ensemble de rôles d'exécution à utiliser sur les clusters qui lui appartiennent. Par exemple, dans une entreprise, chaque service peut utiliser son identifiant externe pour mettre à jour la politique d'approbation relative à son propre ensemble de rôles d'exécution.

Vous pouvez trouver l'ID externe avec l'API Amazon EMR `DescribeSecurityConfiguration`, comme illustré dans l'exemple suivant.

```
aws emr describe-security-configuration --name 'iamconfig-with-lf'{"Name": "iamconfig-
with-lf",
    "SecurityConfiguration":
        "{\"AuthorizationConfiguration\":{\"IAMConfiguration\":
{\"EnableApplicationScopedIAMRole\
        ":true,\"ApplicationScopedIAMRoleConfiguration\":{\"PropagateSourceIdentity
\":true,\"Exter
        nalId\":\"FXH5TSACFDWUCDSR3YQE2O7ETPUSM4OBCGLYWODSCUZDNZ4Y\"}},\"Lake
        FormationConfiguration\":{\"AuthorizedSessionTagValue\":\"Amazon EMR\"}}}",
    "CreationDateTime": "2022-06-03T12:52:35.308000-07:00"
}
```

Pour plus d'informations sur l'utilisation d'un identifiant externe, voir Comment utiliser un identifiant externe lorsque vous accordez l'accès à vos AWS ressources à un tiers.

Audit

Pour surveiller et contrôler les actions que les utilisateurs finaux effectuent avec les rôles IAM, vous pouvez activer la fonctionnalité d'identité source. Pour en savoir plus sur l'identité source, consultez Surveiller et contrôler les actions prises avec les rôles endossés.

Pour suivre l'identité source, `ApplicationScopedIAMRoleConfiguration/PropagateSourceIdentity` `true` configurez-la dans votre configuration de sécurité, comme suit.

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true,
            "ApplicationScopedIAMRoleConfiguration":{
                "PropagateSourceIdentity":true
            }
        }
    }
}
```

Lorsque vous définissez `PropagateSourceIdentity` sur `true`, Amazon EMR applique l'identité source des informations d'identification d'appel à une tâche ou à une session de requête que vous créez avec le rôle d'exécution. Si aucune identité source n'est présente dans les informations d'identification d'appel, Amazon EMR ne définit pas l'identité source.

Pour utiliser cette propriété, accordez des autorisations `sts:SetSourceIdentity` à votre profil d'instance, comme suit.

```
{ // PropagateSourceIdentity statement
    "Sid":"PropagateSourceIdentity",
    "Effect":"Allow",
    "Action":"sts:SetSourceIdentity",
    "Resource":[
        <runtime-role-ARN>
    ],
    "Condition":{
        "StringEquals":{
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

Vous devez également ajouter l'instruction `AllowSetSourceIdentity` à la politique d'approbation de vos rôles d'exécution.

```
{ // AllowSetSourceIdentity statement
    "Sid":"AllowSetSourceIdentity",
```

```
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":[
        "sts:SetSourceIdentity",
        "sts:AssumeRole"
    ],
    "Condition":{
        "StringEquals":{
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

Considérations supplémentaires

> **Note**
>
> Avec la version Amazon EMR`emr-6.9.0`, vous pouvez rencontrer des pannes intermittentes lorsque vous vous connectez à des clusters Amazon EMR depuis Studio. SageMaker Pour résoudre ce problème, vous pouvez installer le correctif avec une action de démarrage lorsque vous lancez le cluster. Pour plus de détails sur le correctif, consultez Problèmes connus de la version 6.9.0 d'Amazon EMR.

Tenez également compte des éléments suivants lorsque vous configurez les rôles d'exécution pour Amazon EMR.

- Amazon EMR prend en charge les rôles d'exécution dans toutes les Régions AWS commerciales.
- Les étapes Amazon EMR prennent en charge les tâches Apache Spark et Apache Hive avec des rôles d'exécution lorsque vous utilisez la version `emr-6.7.0` ou une version ultérieure.
- SageMaker Studio prend en charge les requêtes Spark, Hive et Presto avec des rôles d'exécution lorsque vous utilisez release `emr-6.9.0` ou version ultérieure.
- Les noyaux de bloc-notes suivants SageMaker prennent en charge les rôles d'exécution :
  - DataScience — Noyau Python 3
  - DataScience 2.0 — Noyau Python 3
  - DataScience 3.0 — Noyau Python 3

  - SparkAnalytics 1.0 — SparkMagic et PySpark noyaux
  - SparkAnalytics 2.0 — SparkMagic et PySpark noyaux
  - SparkMagic — PySpark noyau
- Amazon EMR prend en charge les étapes qui utilisent `RunJobFlow` uniquement au moment de la création du cluster. Cette API ne prend pas en charge les rôles d'exécution.
- Amazon EMR ne prend pas en charge les rôles d'exécution sur les clusters que vous configurez pour être hautement disponibles.
- Vous devez échapper à vos arguments de commande Bash lorsque vous exécutez des commandes avec le fichier `command-runner.jar` JAR :

```
aws emr add-steps --cluster-id <cluster-id> --steps '[{"Name":"sample-
step","ActionOnFailure":"CONTINUE","Jar":"command-runner.jar","Properties":"","Args":
["bash","-c","\"aws s3 ls\""],"Type":"CUSTOM_JAR"}]' --execution-role-
arn <IAM_ROLE_ARN>
```

- Les rôles d'exécution ne permettent pas de contrôler l'accès aux ressources du cluster, telles que HDFS et HMS.

Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS .

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 Amazon EMR doit avoir un rôle de service et un rôle pour le profil d'instance Amazon EC2. Pour plus d'informations, consultez Rôle IAM et Utilisation des profils d'instance dans le Guide de l'utilisateur IAM. Les politiques IAM attachées à ces rôles fournissent des autorisations au cluster pour interopérer avec d'autres services AWS pour le compte d'un utilisateur.

Un rôle supplémentaire, le rôle Auto Scaling, est nécessaire si votre cluster utilise la scalabilité automatique dans Amazon EMR. Le rôle AWS de service pour les notebooks EMR est requis si vous utilisez des blocs-notes EMR.

Amazon EMR fournit des rôles par défaut et des politiques gérées par défaut qui déterminent les autorisations pour chaque rôle. Les politiques gérées sont créées et mises à jour par AWS, de sorte qu'elles sont mises à jour automatiquement en cas de modification des exigences de service. Consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

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 Personnaliser les rôles IAM.

Modifier des stratégies basées sur une identité pour autoriser à transmettre des rôles de service pour Amazon EMR

Les politiques gérées par défaut d'Amazon EMR avec autorisations complètes intègrent des configurations de sécurité `iam:PassRole`, notamment les suivantes :

- Les autorisations `iam:PassRole` uniquement pour des rôles Amazon EMR par défaut spécifiques.
- `iam:PassedToService`conditions qui vous permettent d'utiliser la politique uniquement avec AWS des services spécifiques, tels que `elasticmapreduce.amazonaws.com` et`ec2.amazonaws.com`.

Vous pouvez consulter la version JSON des politiques AmazonEMR FullAccess Policy_v2 et AmazonEMR ServicePolicy _v2 dans la console IAM. Nous vous recommandons de créer de nouveaux clusters avec les politiques gérées v2.

Résumé du rôle de service

Le tableau suivant répertorie les rôles de service IAM associés à Amazon EMR pour une référence rapide.

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
Rôle de service pour Amazon EMR (rôle EMR)	`EMR_DefaultRole_V2`	Permet à Amazon EMR d'appeler d'autres AWS services en votre nom lors de la mise en	`AmazonEMRServicePolicy_v2`

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
		service des ressources et de l'exécution d'actions au niveau des services. Ce rôle est obligatoire pour tous les clusters.	**Important** Un rôle lié à un service est nécessaire pour demander des instances Spot. Si ce rôle n'existe pas, le rôle de service Amazon EMR doit avoir l'autorisation de le créer ou une erreur d'autorisation se produit. Si vous prévoyez de demander des instances Spot, vous devez mettre à jour cette politique pour inclure une instruction autorisant la création de ce rôle lié à un service. Pour plus d'informations,

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
			consultez Rôle de service pour Amazon EMR (rôle EMR) la section « Rôle lié au service » pour les demandes d'instance Spot dans le guide de l'utilisateur Amazon EC2.

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
Rôle de service pour les instances EC2 de cluster (profil d'instance EC2)	`EMR_EC2_DefaultRole`	Les processus applicatifs qui s'exécutent au-dessus de l'écosystème Hadoop sur des instances de cluster utilisent ce rôle lorsqu'ils appellent d'autres AWS services. Pour l'accès aux données dans Amazon S3 à l'aide d'EMRFS, vous pouvez spécifier différents rôles à assumer en fonction de l'emplacement des données dans Amazon S3. Par exemple, plusieurs équipes peuvent accéder à un seul « compte de stockage » de données Amazon S3. Pour plus d'informations, consultez Configuration de rôles IAM pour les demandes EMRFS à Amazon S3. Ce rôle est obligatoire pour tous les clusters.	`AmazonElasticMapReduceforEC2Role` . Pour plus d'informations, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
Rôle de service pour le dimensionnement automatique dans Amazon EMR (rôle d'Auto Scaling)	`EMR_AutoScaling_DefaultRole`	Permet des 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 de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.	`AmazonElasticMapReduceforAutoScalingRole` . Pour plus d'informations, consultez Rôle de service pour le dimensionnement automatique dans Amazon EMR (rôle d'Auto Scaling).

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
Rôle de service pour Blocs-notes EMR	`EMR_Notebooks_DefaultRole`	Fournit les autorisations dont un bloc-notes EMR a besoin pour accéder à d'autres AWS ressources et effectuer des actions. Nécessaire uniquement si Blocs-notes EMR sont utilisés.	`AmazonElasticMapReduceEditorsRole` . Pour plus d'informations, consultez Rôle de service pour Blocs-notes EMR. `S3FullAccessPolicy` est également attaché par défaut. Voici le contenu de cette politique. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:*", "Resource": "*" } ] }`

Fonction	Rôle par défaut	Description	Stratégie gérée par défaut
Rôle lié à un service	`AWSServiceRoleForEMRCleanup`	Amazon EMR créé automatiquement un rôle lié à un service. Si le service pour Amazon EMR a perdu la capacité de nettoyer les ressources Amazon EC2, Amazon EMR peut utiliser ce rôle pour effectuer le nettoyage. Si un cluster utilise des instances Spot, la stratégie d'autorisation attachée au Rôle de service pour Amazon EMR (rôle EMR) doit autoriser la création d'un rôle lié à un service. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Amazon EMR.	`AmazonEMRCleanupPolicy`

Rubriques

- Rôles de service IAM utilisés par Amazon EMR
- Personnaliser les rôles IAM
- Configuration de rôles IAM pour les demandes EMRFS à Amazon S3
- Utilisation de politiques basées sur les ressources pour l'accès d'Amazon EMR au catalogue de données AWS Glue

- Utilisation des rôles IAM avec des applications qui appellent directement les services AWS
- Permettre aux utilisateurs et aux groupes de créer et de modifier des rôles

Rôles de service IAM utilisés par Amazon EMR

Amazon EMR utilise les rôles de service IAM pour effectuer des actions en votre nom lors du provisionnement des ressources du cluster, de l'exécution des applications, de la mise à l'échelle dynamique des ressources et de la création et de l'exécution de Blocs-notes EMR. Amazon EMR utilise les rôles suivants lorsqu'il interagit avec d'autres services AWS . Chaque rôle a une fonction unique au sein d'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 le code d'application de votre cluster appelle directement les AWS services, vous devrez peut-être utiliser le SDK pour spécifier les rôles. Pour plus d’informations, consultez Utilisation des rôles IAM avec des applications qui appellent directement les services AWS.

Rubriques

- Rôle de service pour Amazon EMR (rôle EMR)
- Rôle de service pour les instances EC2 de cluster (profil d'instance EC2)
- Rôle de service pour le dimensionnement automatique dans Amazon EMR (rôle d'Auto Scaling)
- Rôle de service pour Blocs-notes EMR
- Utilisation de rôles liés à un service pour Amazon EMR

Rôle de service pour Amazon EMR (rôle EMR)

Le rôle Amazon EMR définit les actions autorisées pour Amazon EMR lorsqu'il alloue des ressources et exécute des tâches de niveau service qui ne sont pas effectuées dans le contexte d'une instance Amazon EC2 exécutée au sein d'un cluster. Par exemple, le rôle de service est utilisé pour mettre en service des instances EC2 lorsqu'un cluster est lancé.

- Le nom de rôle par défaut est `EMR_DefaultRole_V2`.
- La politique gérée par défaut limitée à Amazon EMR attachée à `EMR_DefaultRole_V2` est `AmazonEMRServicePolicy_v2`. Cette politique v2 remplace la politique gérée par défaut obsolète, `AmazonElasticMapReduceRole`.

`AmazonEMRServicePolicy_v2` dépend de l'accès limité aux ressources qu'Amazon EMR fournit ou utilise. Lorsque vous utilisez cette politique, vous devez transmettre la balise utilisateur `for-use-with-amazon-emr-managed-policies = true` lors du provisionnement du cluster. Amazon EMR propagera automatiquement ces balises. Vous pouvez également avoir besoin d'ajouter manuellement une balise utilisateur à des types de ressources spécifiques, tels que les groupes de sécurité EC2 qui n'ont pas été créés par Amazon EMR. veuillez consulter Balisage des ressources pour l'utilisation des politiques gérées.

> **Important**
>
> Amazon EMR utilise cette fonction du service Amazon EMR et le rôle `AWSServiceRoleForEMRCleanup` pour nettoyer les ressources de cluster de votre compte que vous n'utilisez plus, telles que les instances Amazon EC2. Vous devez inclure des actions pour les politiques de rôle afin de supprimer ou de résilier les ressources. Dans le cas contraire, Amazon EMR ne pourra pas effectuer ces actions de nettoyage, et les ressources non utilisées qui restent sur le cluster risquent de générer des coûts.

L'illustration suivante montre le contenu de la politique `AmazonEMRServicePolicy_v2` actuelle. Vous pouvez également voir le contenu actuel de la politique gérée `AmazonEMRServicePolicy_v2` sur la console IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CreateInTaggedNetwork",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:RunInstances",
    "ec2:CreateFleet",
    "ec2:CreateLaunchTemplate",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
```

```
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateWithEMRTaggedLaunchTemplate",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateFleet",
    "ec2:RunInstances",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateEMRTaggedLaunchTemplate",
   "Effect": "Allow",
   "Action": "ec2:CreateLaunchTemplate",
   "Resource": "arn:aws:ec2:*:*:launch-template/*",
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateEMRTaggedInstancesAndVolumes",
   "Effect": "Allow",
   "Action": [
    "ec2:RunInstances",
    "ec2:CreateFleet"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:volume/*"
   ],
   "Condition": {
    "StringEquals": {
```

```
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "ResourcesToLaunchEC2",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances",
     "ec2:CreateFleet",
     "ec2:CreateLaunchTemplate",
     "ec2:CreateLaunchTemplateVersion"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*::image/ami-*",
     "arn:aws:ec2:*:*:key-pair/*",
     "arn:aws:ec2:*:*:capacity-reservation/*",
     "arn:aws:ec2:*:*:placement-group/pg-*",
     "arn:aws:ec2:*:*:fleet/*",
     "arn:aws:ec2:*:*:dedicated-host/*",
     "arn:aws:resource-groups:*:*:group/*"
    ]
   },
   {
    "Sid": "ManageEMRTaggedResources",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateLaunchTemplateVersion",
     "ec2:DeleteLaunchTemplate",
     "ec2:DeleteNetworkInterface",
     "ec2:ModifyInstanceAttribute",
     "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "ManageTagsOnEMRTaggedResources",
    "Effect": "Allow",
```

```
    "Action": [
     "ec2:CreateTags",
     "ec2:DeleteTags"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateNetworkInterfaceNeededForPrivateSubnet",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateNetworkInterface"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "TagOnCreateTaggedEMRResources",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*",
     "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
```

```
    "StringEquals": {
     "ec2:CreateAction": [
      "RunInstances",
      "CreateFleet",
      "CreateLaunchTemplate",
      "CreateNetworkInterface"
     ]
    }
   }
  },
  {
   "Sid": "TagPlacementGroups",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateTags",
    "ec2:DeleteTags"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:placement-group/pg-*"
   ]
  },
  {
   "Sid": "ListActionsForEC2Resources",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceTypeOfferings",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribePlacementGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:DescribeVolumeStatus",
    "ec2:DescribeVpcAttribute",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcs"
   ],
```

```
    "Resource": "*"
   },
   {
    "Sid": "CreateDefaultSecurityGroupWithEMRTags",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSecurityGroup"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:security-group/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateDefaultSecurityGroupInVPCWithEMRTags",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateSecurityGroup"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:vpc/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "TagOnCreateDefaultSecurityGroupWithEMRTags",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
      "ec2:CreateAction": "CreateSecurityGroup"
     }
```

```
    }
   },
   {
    "Sid": "ManageSecurityGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:AuthorizeSecurityGroupEgress",
     "ec2:AuthorizeSecurityGroupIngress",
     "ec2:RevokeSecurityGroupEgress",
     "ec2:RevokeSecurityGroupIngress"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateEMRPlacementGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:CreatePlacementGroup"
    ],
    "Resource": "arn:aws:ec2:*:*:placement-group/pg-*"
   },
   {
    "Sid": "DeletePlacementGroups",
    "Effect": "Allow",
    "Action": [
     "ec2:DeletePlacementGroup"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AutoScaling",
    "Effect": "Allow",
    "Action": [
     "application-autoscaling:DeleteScalingPolicy",
     "application-autoscaling:DeregisterScalableTarget",
     "application-autoscaling:DescribeScalableTargets",
     "application-autoscaling:DescribeScalingPolicies",
     "application-autoscaling:PutScalingPolicy",
     "application-autoscaling:RegisterScalableTarget"
```

```
    ],
    "Resource": "*"
   },
   {
    "Sid": "ResourceGroupsForCapacityReservations",
    "Effect": "Allow",
    "Action": [
     "resource-groups:ListGroupResources"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AutoScalingCloudWatch",
    "Effect": "Allow",
    "Action": [
     "cloudwatch:PutMetricAlarm",
     "cloudwatch:DeleteAlarms",
     "cloudwatch:DescribeAlarms"
    ],
    "Resource": "arn:aws:cloudwatch:*:*:alarm:*_EMR_Auto_Scaling"
   },
   {
    "Sid": "PassRoleForAutoScaling",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
    "Condition": {
     "StringLike": {
      "iam:PassedToService": "application-autoscaling.amazonaws.com*"
     }
    }
   },
   {
    "Sid": "PassRoleForEC2",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
    "Condition": {
     "StringLike": {
      "iam:PassedToService": "ec2.amazonaws.com*"
     }
    }
   }
  ]
```

```
}
```

Votre rôle de service doit utiliser la politique d'approbation suivante.

> **Important**
>
> La politique d'approbation suivante inclut les clés de condition globales `aws:SourceArn` et `aws:SourceAccount`, qui limitent les autorisations que vous accordez à Amazon EMR pour des ressources particulières de votre compte. L'utilisation de ces clés peut vous protéger contre le problème de l'adjoint confus.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

Rôle de service pour les instances EC2 de cluster (profil d'instance EC2)

Le rôle de service pour les instances EC2 de cluster (également appelé profil d'instance EC2 pour Amazon EMR) est un type spécial de rôle de service qui est attribué 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 Guide de l'utilisateur IAM.

> **Important**
>
> Le rôle de service par défaut pour les instances EC2 du cluster et la politique gérée AWS par défaut associée `AmazonElasticMapReduceforEC2Role` sont sur le point de devenir obsolètes, aucune politique AWS gérée de remplacement n'étant fournie. Vous devrez créer et spécifier un profil d'instance pour remplacer le rôle obsolète et la politique par défaut.

Rôle et stratégie gérée par défaut

- Le nom de rôle par défaut est `EMR_EC2_DefaultRole`.
- La politique gérée par `EMR_EC2_DefaultRole` par défaut, `AmazonElasticMapReduceforEC2Role`, arrive en fin de support. Au lieu d'utiliser une politique gérée par défaut pour le profil d'instance EC2, appliquez des politiques basées sur les ressources aux compartiments S3 et aux autres ressources dont Amazon EMR a besoin, ou utilisez votre propre politique gérée par le client avec un rôle IAM comme profil d'instance. Pour plus d'informations, consultez Création d'un rôle de service pour les instances EC2 de cluster avec des autorisations de moindre privilège.

Vous trouverez ci-dessous le contenu de la version 3 de `AmazonElasticMapReduceforEC2Role`.

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

Votre rôle de service doit utiliser la politique d'approbation suivante.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Création d'un rôle de service pour les instances EC2 de cluster avec des autorisations de moindre privilège

À titre de bonne pratique, nous vous recommandons vivement de créer un rôle de service pour les instances EC2 du cluster et une politique d'autorisations qui prévoit les autorisations minimales pour les autres AWS services requis par votre application.

La stratégie gérée par défaut, `AmazonElasticMapReduceforEC2Role`, fournit des autorisations qui facilitent le lancement d'un cluster initial. Cependant, `AmazonElasticMapReduceforEC2Role` il est sur le point de devenir obsolète et Amazon EMR ne fournira pas de politique AWS gérée par défaut de remplacement pour le rôle obsolète. Pour lancer un cluster initial, vous devez fournir une politique basée sur les ressources ou sur les identifiants gérée par le client.

Les déclarations de politique suivantes fournissent des exemples d'autorisations requises pour différentes fonctionnalités 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 exemples de déclarations de politique utilisent la *`us-west-2`*région et l'identifiant de AWS compte fictif *`123456789012`*. Remplacez-les par les bonnes informations pour votre cluster.

Pour plus d'informations sur la création et la spécification des rôles personnalisés, consultez Personnaliser les rôles IAM.

> **Note**
>
> Si vous créez un rôle EMR personnalisé 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 avec des noms différents, mais Amazon EMR ne prend pas en charge cette configuration, et il en résulte une erreur « profil d'instance non valide » lorsque vous créez le cluster.

Lecture et écriture de données sur Amazon S3 à l'aide d'EMRFS

Lorsqu'une application exécutée sur un cluster Amazon EMR référence des données à l'aide du format `s3://mydata`, Amazon EMR utilise le profil d'instance EC2 pour effectuer la demande. Les clusters lisent et écrivent généralement des données sur Amazon S3 de cette manière, et Amazon EMR utilise par défaut les autorisations attachées au rôle de service pour les instances EC2 de cluster. Pour plus d'informations, consultez Configuration de rôles IAM pour les demandes EMRFS à Amazon S3.

Étant donné que les rôles IAM pour EMRFS basculeront vers les autorisations attachées au rôle de service pour les instances EC2 de cluster, nous vous recommandons, à titre de bonne pratique, d'utiliser les rôles IAM pour EMRFS et de limiter les autorisations EMRFS et 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* spécifie le compartiment Amazon S3 dans lequel le cluster lit et écrit les données et tous les sous-dossiers utilisant */**. Ajoutez uniquement les compartiments et les dossiers dont votre application a besoin.
- La déclaration de politique qui autorise les actions dynamodb n'est requise que si la vue cohérente EMRFS est activée. *EmrFSMetadata* spécifie le dossier par défaut pour la vue cohérente EMRFS.

```
{
    "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:ListBucket"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:DeleteQueue",
                "sqs:SendMessage",
                "sqs:CreateQueue"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:EMRFS-Inconsistency-*"
        }
    ]
}
```

Archivage des fichiers journaux sur Amazon S3

La déclaration de politique suivante permet au cluster Amazon EMR d'archiver les fichiers journaux vers l'emplacement Amazon S3 spécifié. Dans l'exemple ci-dessous, la date de création du cluster a *s3://MyLoggingBucket/MyEMRClusterLogs*été spécifiée à l'aide de l'emplacement S3 du dossier Log dans la console, à l'aide de l'`--log-uri` AWS CLI option du ou à l'aide du `LogUri` paramètre de la `RunJobFlow` commande. Pour plus d'informations, consultez Archiver les fichiers journaux sur Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::MyLoggingBucket/MyEMRClusterLogs/*"
        }
    ]
}
```

Utilisation des outils de débogage

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 Activation de l'outil de débogage.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:AWS-ElasticMapReduce-*"
        }
    ]
}
```

Utilisation du catalogue AWS de données Glue

La déclaration de politique suivante autorise les actions requises si vous utilisez le AWS Glue Data Catalog comme métastore pour les applications. Pour plus d'informations, consultez les sections Utilisation du catalogue de données AWS Glue comme métastore pour Spark SQL, Utilisation du catalogue de données AWS Glue comme métastore pour Hive et Utilisation de Presto avec le catalogue de AWS données Glue dans le guide de publication d'Amazon EMR.

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

Rôle de service pour le dimensionnement automatique dans Amazon EMR (rôle d'Auto Scaling)

Le rôle d'Auto Scaling pour Amazon EMR a une fonction similaire au rôle de service, mais permet des actions supplémentaires pour les environnements au dimensionnement dynamique.

- Le nom de rôle par défaut est `EMR_AutoScaling_DefaultRole`.
- La stratégie gérée par défaut attachée à `EMR_AutoScaling_DefaultRole` est `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": "*"
```

```
        }
    ]
}
```

Votre rôle de service doit utiliser la politique d'approbation suivante.

> **Important**
>
> La politique d'approbation suivante inclut les clés de condition globales `aws:SourceArn` et `aws:SourceAccount`, qui limitent les autorisations que vous accordez à Amazon EMR pour des ressources particulières de votre compte. L'utilisation de ces clés peut vous protéger contre le problème de l'adjoint confus.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "application-autoscaling.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:application-
autoscaling:<region>:<account-id>:scalable-target/*"
                }
            }
        }
    ]
}
```

Rôle de service pour Blocs-notes EMR

Chaque bloc-notes EMR a besoin d'autorisations pour accéder à d'autres AWS ressources et effectuer des actions. Les politiques IAM associées à ce rôle de service autorisent le bloc-notes à interagir avec d'autres AWS services. Lorsque vous créez un bloc-notes à l'aide du AWS

Management Console, vous spécifiez un rôle AWS de service. Vous pouvez utiliser le rôle par défaut, `EMR_Notebooks_DefaultRole`, ou spécifier un rôle que vous créez. Si un bloc-notes n'a pas été créé auparavant, vous pouvez choisir de créer le rôle par défaut.

- Le nom de rôle par défaut est `EMR_Notebooks_DefaultRole`.
- Les politiques gérées par défaut attachées à `EMR_Notebooks_DefaultRole` sont `AmazonElasticMapReduceEditorsRole` et `S3FullAccessPolicy`.

Votre rôle de service doit utiliser la politique d'approbation suivante.

> **Important**
>
> La politique d'approbation suivante inclut les clés de condition globales `aws:SourceArn` et `aws:SourceAccount`, qui limitent les autorisations que vous accordez à Amazon EMR pour des ressources particulières de votre compte. L'utilisation de ces clés peut vous protéger contre le problème de l'adjoint confus.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

Le contenu de la version 1 de `AmazonElasticMapReduceEditorsRole` est le suivant.

```
{
    "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",
                "ec2:DescribeVpcs",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListSteps"
            ],
            "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"
                    ]
                }
            }
        }
    ]
}
```

Voici le contenu de `S3FullAccessPolicy`. `S3FullAccessPolicy` permet à votre rôle de service pour Blocs-notes EMR d'effectuer toutes les actions Amazon S3 sur les objets de votre Compte AWS. Lorsque vous créez un rôle de service personnalisé pour Blocs-notes EMR, vous devez lui donner des autorisations Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
    ]
}
```

Vous pouvez limiter l'accès en lecture et en écriture de votre rôle de service à l'emplacement Amazon S3 où vous voulez enregistrer les fichiers de vos blocs-notes. Utilisez l'ensemble minimum d'autorisations Amazon S3 suivant.

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

Si votre compartiment Amazon S3 est chiffré, vous devez inclure les autorisations suivantes pour AWS Key Management Service.

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

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 politique IAM attachée à la fonction du service pour les blocs-notes Amazon EMR. Voici un exemple de stratégie :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

Autorisations du rôle du service de Blocs-notes EMR

Ce tableau répertorie les actions que les blocs-notes EMR effectuent à l'aide du rôle de service, ainsi que les autorisations nécessaires pour chaque action.

Action	Autorisations
Établir un canal réseau sécurisé entre un bloc-notes et un cluster Amazon EMR, et effectuer les actions de nettoyage nécessaires.	`"ec2:CreateNetworkInterface", "ec2:CreateNetworkInterfacePermission", "ec2:DeleteNetworkInterface", "ec2:DeleteNetworkInterfacePermission", "ec2:DescribeNetworkInterfaces", "ec2:ModifyNetworkInterfaceAttribute", "ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2:CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce:ListInstances", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListSteps"`
Utiliser les informations d'identification Git stockées dans AWS	`"secretsmanager:GetSecretValue"`

Action	Autorisations
Secrets Manager pour lier des référentiels Git à un bloc-notes.	
Appliquez des AWS balises à l'interface réseau et aux groupes de sécurité par défaut créés par EMR Notebooks lors de la configuration du canal réseau sécurisé. Pour plus d'informations, veuillez consulter la rubrique Balisage des ressources AWS.	`"ec2:CreateTags"`

Accédez aux fichiers et aux métadonnées des blocs-notes ou chargez-les sur Amazon S3.	`"s3:PutObject",`
	`"s3:GetObject",`
	`"s3:GetEncryptionConfiguration",`
	`"s3:ListBucket",`
	`"s3:DeleteObject"`
	
	Les autorisations suivantes ne sont requises que si vous utilisez un compartiment Amazon S3 chiffré.
	
	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`
	`"kms:ReEncryptFrom",`
	`"kms:ReEncryptTo",`
	`"kms:DescribeKey"`

Mises à jour des politiques gérées par EMR Notebooks AWS

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour les EMR Notebooks depuis le 1er mars 2021.

Modification	Description	Date
`AmazonElasticMapReduceEditorsRole - Added permissions`	Les blocs-notes EMR ont ajouté les autorisations `ec2:describeVPCs` et	8 février 2023

Modification	Description	Date
	`elastmicmapreduce: ListSteps` à `AmazonElasticMapReduceEditorsRole` .	
Les blocs-notes EMR ont commencé à suivre les modifications.	EMR Notebooks a commencé à suivre les modifications apportées à ses politiques gérées. AWS	8 février 2023

Utilisation de rôles liés à un service pour Amazon EMR

Amazon EMR utilise des rôles liés à un AWS Identity and Access Management service (IAM). Un rôle lié à un service est un type unique de rôle IAM qui est lié directement à Amazon EMR. Les rôles liés à un service sont prédéfinis par Amazon EMR et incluent toutes les autorisations requises par le service pour appeler d'autres AWS services en votre nom.

Rubriques

- Utilisation de rôles liés à un service pour le nettoyage
- Utilisation de rôles liés à un service pour la journalisation anticipée

Pour plus d’informations sur les autres services prenant en charge les rôles liés à un service, consultez les AWS services opérationnels avec IAM et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Utilisation de rôles liés à un service pour le nettoyage

Amazon EMR utilise des rôles liés à un AWS Identity and Access Management service (IAM). Un rôle lié à un service est un type unique de rôle IAM qui est lié directement à Amazon EMR. Les rôles liés à un service sont prédéfinis par Amazon EMR et incluent toutes les autorisations requises par le service pour appeler d'autres AWS services en votre nom.

Les rôles liés à un service fonctionnent conjointement avec le rôle de 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 Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..

Un rôle lié à un service facilite la configuration d'Amazon EMR, car vous n'avez pas à ajouter manuellement les autorisations nécessaires. Amazon EMR définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul Amazon EMR peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous ne pouvez supprimer ce rôle lié à un service pour Amazon EMR qu'après avoir supprimé les ressources associées et résilié tous les clusters EMR du compte. Cela protège vos ressources Amazon EMR afin que vous ne puissiez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Utilisation de rôles liés à un service pour le nettoyage

Amazon EMR utilise le AWSServiceRoleForEMRCleanuprôle basé sur le service pour autoriser Amazon EMR à résilier et à supprimer les ressources Amazon EC2 en votre nom si le rôle lié au service Amazon EMR perd cette fonctionnalité. Amazon EMR crée automatiquement le rôle lié au service lors de la création du cluster s'il n'existe pas déjà.

Le rôle AWSServiceRoleForEMRCleanup lié à un service fait confiance aux services suivants pour assumer le rôle :

- `elasticmapreduce.amazonaws.com`

La politique d'autorisation des rôles AWSServiceRoleForEMRCleanup liés au service permet à Amazon EMR d'effectuer 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, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service.

Création d'un rôle lié à un service pour Amazon EMR

Il n'est pas nécessaire de créer le AWSServiceRoleForEMRCleanup rôle manuellement. Lorsque vous lancez un cluster, que ce soit pour la première fois ou lorsque le rôle AWSServiceRoleForEMRCleanup lié au service n'est pas présent, Amazon EMR crée le rôle lié au AWSServiceRoleForEMRCleanup service pour vous. Vous devez disposer des autorisations nécessaires pour créer un rôle lié à un 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 Utilisation de rôles liés à un service pour le nettoyage.

> **Important**
>
> Si vous avez utilisé Amazon EMR avant le 24 octobre 2017, date à laquelle les rôles liés à un service n'étaient pas pris en charge, Amazon EMR a créé le AWSServiceRoleForEMRCleanup rôle lié à un service 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 au AWSServiceRoleForEMRCleanup service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas modifier le nom du rôle lié à un service car diverses entités peuvent faire référence au rôle lié à un service. Toutefois, vous pouvez modifier la description du rôle lié au service à l'aide d'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 panneau de navigation de la console IAM, sélectionnez Roles (Rôles).

2. Choisissez le nom du rôle à modifier.
3. À droite de Description du rôle, choisissez Modifier.
4. Saisissez une nouvelle description dans le champ et sélectionnez Save changes (Enregistrer les modifications).

Modification de la description d'un rôle lié à un service (CLI IAM)

Vous pouvez utiliser les commandes IAM depuis le 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 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 : description UpdateRole

Suppression d'un rôle lié à un service pour Amazon EMR

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service nécessitant un rôle lié à un service, nous vous recommandons de supprimer ce rôle lié à un service. 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.

Nettoyer un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord confirmer que le rôle lié à un service n'a aucune session active et supprimer toutes les ressources utilisées par le rôle lié à un service.

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, sélectionnez Rôles. Sélectionnez le nom (et non la case à cocher) du rôle AWSServiceRoleForEMRCleanup lié au service.
3. Sur la page Résumé du rôle lié au service sélectionné, choisissez Access Advisor.
4. Dans l'onglet Access Advisor, consultez l'activité récente pour le rôle lié à un service.

> **Note**
>
> Si vous ne savez pas si Amazon EMR utilise le rôle AWSServiceRoleForEMRCleanup lié au service, vous pouvez essayer de le supprimer. Si le service utilise le rôle lié au service, la suppression échoue et vous pouvez afficher les régions dans lesquelles le rôle lié au service est utilisé. Si le rôle lié au service est utilisé, vous devez attendre la fin de la session avant de pouvoir supprimer le rôle lié au service. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Pour supprimer les ressources Amazon EMR utilisées par AWSServiceRoleForEMRCleanup

- Mettez fin à tous les clusters de votre compte. Pour plus d’informations, consultez Arrêter un cluster.

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, sélectionnez Rôles. Cochez la case située à côté AWSServiceRoleForEMRCleanup, et non le nom ou la ligne elle-même.
3. Pour les actions sur les Rôle en haut de la page, sélectionnez Supprimer.
4. Dans la boîte de dialogue de confirmation, passez en revue les données du dernier accès au service, qui indiquent la date à laquelle chacun des rôles sélectionnés a accédé à un AWS service pour la dernière fois. 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. La suppression du rôle lié au service IAM étant asynchrone, une fois que vous avez soumis le rôle lié au service pour suppression, la tâche de suppression peut réussir ou échouer. Si la tâche échoue, vous pouvez choisir View details (Afficher les détails) ou View Resources (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 (CLI IAM)

Vous pouvez utiliser les commandes IAM depuis le 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 soumettre une demande de suppression pour un rôle lié à un service, appelez. DeleteServiceLinkedRole Dans la demande, spécifiez le nom du AWSServiceRoleForEMRCleanup rôle.

   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 le `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 AWSServiceRoleForEMRCleanup

Amazon EMR prend en charge l'utilisation du rôle AWSServiceRoleForEMRCleanup lié au service dans les régions suivantes.

Nom de la région	Identité de la région	Prise en charge dans Amazon EMR
US East (Virginie du Nord)	us-east-1	Oui
USA Est (Ohio)	us-east-2	Oui

Nom de la région	Identité de la région	Prise en charge dans Amazon EMR
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)	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

Utilisation de rôles liés à un service pour la journalisation anticipée

Amazon EMR utilise des rôles liés à un AWS Identity and Access Management service (IAM). Un rôle lié à un service est un type unique de rôle IAM qui est lié directement à Amazon EMR. Les rôles liés à un service sont prédéfinis par Amazon EMR et incluent toutes les autorisations requises par le service pour appeler d'autres AWS services en votre nom.

Les rôles liés à un service fonctionnent conjointement avec le rôle de 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 Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..

Un rôle lié à un service facilite la configuration d'Amazon EMR, car vous n'avez pas à ajouter manuellement les autorisations nécessaires. Amazon EMR définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul Amazon EMR peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous ne pouvez supprimer ce rôle lié à un service pour Amazon EMR qu'après avoir supprimé les ressources associées et résilié tous les clusters EMR du compte. Cela protège vos ressources Amazon EMR afin que vous ne puissiez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Autorisations de rôle liées au service pour la journalisation en écriture anticipée (WAL)

Amazon EMR utilise le rôle lié au service AWSServiceRoleForEMRWALpour récupérer l'état d'un cluster.

Le rôle AWSServiceRoleForEMRWAL lié à un service fait confiance aux services suivants pour assumer le rôle :

- `emrwal.amazonaws.com`

La politique `EMRDescribeClusterPolicyForEMRWAL`d'autorisations pour le rôle lié à un service permet à Amazon EMR d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `DescribeCluster` sur *

Vous devez configurer les autorisations pour autoriser une entité IAM (dans ce cas, Amazon EMR WAL) à créer, modifier ou supprimer un rôle lié à un service. Ajoutez les instructions suivantes, le cas échéant, à la politique d'autorisation de votre profil d'instance :

CreateServiceLinkedRole

Pour autoriser une entité IAM à créer le rôle lié à un AWSServiceRoleForEMRWAL service

Ajoutez l'instruction suivante à la stratégie d'autorisation 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/emrwal.amazonaws.com*/
AWSServiceRoleForEMRWAL*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "emrwal.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

UpdateRoleDescription

Pour autoriser une entité IAM à modifier la description du rôle lié à un AWSServiceRoleForEMRWAL service

Ajoutez l'instruction suivante à la stratégie d'autorisation 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/emrwal.amazonaws.com*/
AWSServiceRoleForEMRWAL*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "emrwal.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

DeleteServiceLinkedRole

Pour autoriser une entité IAM à supprimer le rôle lié à un AWSServiceRoleForEMRWAL service

Ajoutez l'instruction suivante à la stratégie d'autorisation 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": [
                "emrwal.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Création d'un rôle lié à un service pour Amazon EMR

Il n'est pas nécessaire de créer le AWSServiceRoleForEMRWAL rôle manuellement. Amazon EMR crée automatiquement ce rôle lié à un service lorsque vous créez un espace de travail WAL avec la CLI EMRWAL ou depuis AWS CloudFormation, ou HBase créera le rôle lié au service lorsque vous configurez un espace de travail pour Amazon EMR WAL et que le rôle lié au service n'existe pas encore. Vous devez disposer des autorisations nécessaires pour créer un rôle lié à un service. Pour des exemples d'instructions qui ajoutent cette fonctionnalité à la politique d'autorisations d'une entité IAM (telle qu'un utilisateur, un groupe ou un rôle), consultez la section précédente,Autorisations de rôle liées au service pour la journalisation en écriture anticipée (WAL).

Modification d'un rôle lié à un service pour Amazon EMR

Amazon EMR ne vous permet pas de modifier le rôle lié au AWSServiceRoleForEMRWAL service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas modifier le nom du rôle lié à un service car diverses entités peuvent faire référence au rôle lié à un service. Toutefois, vous pouvez modifier la description du rôle lié au service à l'aide d'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 panneau de navigation de la console IAM, sélectionnez Roles (Rôles).
2. Choisissez le nom du rôle à modifier.
3. À droite de Description du rôle, choisissez Modifier.
4. Saisissez une nouvelle description dans le champ et sélectionnez Save changes (Enregistrer les modifications).

Modification de la description d'un rôle lié à un service (CLI IAM)

Vous pouvez utiliser les commandes IAM depuis le 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 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 : description UpdateRole

Suppression d'un rôle lié à un service pour Amazon EMR

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service nécessitant un rôle lié à un service, nous vous recommandons de supprimer ce rôle lié à un service. 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.

> **Note**
>
> L'opération de journalisation par écriture anticipée n'est pas affectée si vous supprimez le AWSServiceRoleForEMRWAL rôle, mais Amazon EMR ne supprimera pas automatiquement les journaux créés une fois votre cluster EMR terminé. Par conséquent, vous devrez supprimer manuellement les journaux Amazon EMR WAL si vous supprimez le rôle lié au service.

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, sélectionnez Rôles. Sélectionnez le nom (et non la case à cocher) du AWSServiceRoleForEMRWAL rôle.
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 ne savez pas si Amazon EMR utilise le AWSServiceRoleForEMRWAL rôle, vous pouvez essayer de supprimer le rôle lié au service. Si le service utilise le rôle, la

suppression échoue et vous pouvez afficher les régions dans lesquelles le rôle lié au service est utilisé. Si le rôle lié au service est utilisé, vous devez attendre la fin de la session avant de pouvoir supprimer le rôle lié au service. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Pour supprimer les ressources Amazon EMR utilisées par AWSServiceRoleForEMRWAL

- Mettez fin à tous les clusters de votre compte. Pour plus d'informations, consultez Arrêter un cluster.

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, sélectionnez Rôles. Cochez la case située à côté AWSServiceRoleForEMRWAL, et non le nom ou la ligne elle-même.
3. Pour les actions sur les Rôle en haut de la page, sélectionnez Supprimer.
4. Dans la boîte de dialogue de confirmation, passez en revue les données du dernier accès au service, qui indiquent la date à laquelle chacun des rôles sélectionnés a accédé à un AWS service pour la dernière fois. 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 View details (Afficher les détails) ou View Resources (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 (CLI IAM)

Vous pouvez utiliser les commandes IAM depuis le 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 AWSServiceRoleForEMRWAL
   ```

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 soumettre une demande de suppression pour un rôle lié à un service, appelez. DeleteServiceLinkedRole Dans la demande, spécifiez le nom du AWSServiceRoleForEMRWAL rôle.

   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 le `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 AWSServiceRoleForEMRWAL

Amazon EMR prend en charge l'utilisation du rôle AWSServiceRoleForEMRWAL lié au service dans les régions suivantes.

Nom de la région	Identité de la région	Prise en charge dans Amazon EMR
US East (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 (Singapour)	ap-southeast-1	Oui
Asie-Pacifique (Sydney)	ap-southeast-2	Oui
Asie-Pacifique (Tokyo)	ap-northeast-1	Oui
Europe (Francfort)	eu-central-1	Oui
Europe (Irlande)	eu-west-1	Oui

Personnaliser les rôles IAM

Vous avez la possibilité de personnaliser les fonctions du 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 Permettre aux utilisateurs et aux groupes de créer et de modifier des rôles.

Si vous créez un rôle EMR personnalisé 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 avec des noms différents, mais Amazon EMR ne prend pas en charge cette configuration, et il en résulte une erreur « 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 politiques en ligne, vous devez savoir que des mises à jour de service peuvent se produire et provoquer soudainement des erreurs d'autorisation. Pour plus d'informations, consultez Stratégies gérées et stratégies en ligne dans le Guide de l'utilisateur IAM et Spécifiez les rôles IAM personnalisés lors de la création d'un cluster.

Pour plus d'informations sur la manipulation de rôles IAM, consultez les rubriques suivantes dans le Guide de l'utilisateur IAM :

- Création d'un rôle pour déléguer des autorisations à un AWS service
- Modification d'un rôle
- Suppression d'un rôle

Spécifiez les rôles IAM personnalisés lors de la création d'un 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 n'est pas autorisé à appeler EC2 se produit. Pour plus d'informations, consultez Permettre aux utilisateurs et aux groupes de créer et de modifier des rôles.

Utiliser la console pour spécifier des rôles personnalisés

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 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 Rôles de service IAM utilisés par Amazon EMR.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Pour spécifier des rôles IAM personnalisés avec la nouvelle console

Vous spécifiez le rôle de service pour Amazon EMR et le rôle pour le profil d'instance Amazon EC2 lorsque vous créez un cluster. Pour plus d'informations, consultez Rôles de service IAM utilisés par Amazon EMR.

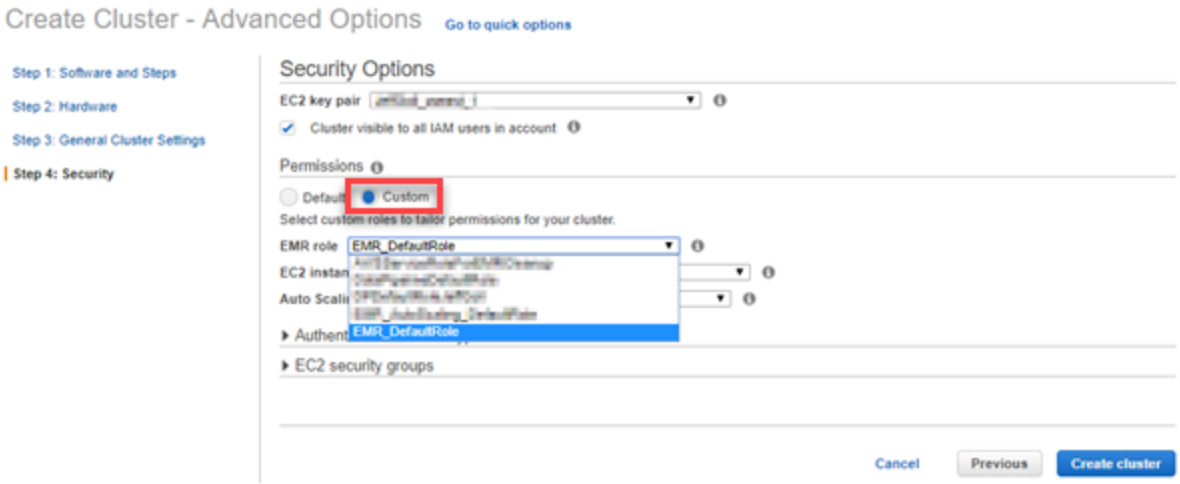
1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Configuration de la sécurité et autorisations, recherchez les champs Rôle IAM pour le profil d'instance et Rôle de service pour Amazon EMR. 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.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour spécifier des rôles IAM personnalisés avec l'ancienne console

Lorsque vous créez un cluster avec l'ancienne console, vous pouvez spécifier un rôle de service personnalisé pour Amazon EMR, un rôle personnalisé pour le profil d'instance EC2 et un rôle personnalisé Auto Scaling à l'aide des 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 Rôles de service IAM utilisés par Amazon EMR.

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster et Go to advanced options (Aller aux options avancées).
3. Choisissez les réglages de cluster appropriés pour votre application jusqu'à ce que vous atteigniez les Options de sécurité. Sous 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.

Utilisez le AWS CLI pour spécifier des rôles personnalisés

Vous pouvez spécifier un rôle de service pour Amazon EMR et un rôle de service pour les instances EC2 du cluster en utilisant explicitement des options avec la commande `create-cluster` de 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 rôle Auto Scaling est spécifié à l'aide d'une option séparée `--auto-scaling-role`. Pour plus d'informations, consultez Utilisation de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.

Pour spécifier des rôles IAM personnalisés à l'aide du 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 continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.1.0 \
--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 illustre le contenu d'un `config` fichier pour AWS CLI les rôles personnalisés spécifiés pour Amazon EMR. Avec ce fichier de configuration, lorsque l'option `--use-default-roles` est spécifiée, le cluster est créé en utilisant les *MyCustomServiceRoleForEMR*

et *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
```

Configuration de rôles IAM pour les demandes EMRFS à Amazon S3

> **Note**
>
> La fonctionnalité de mappage de rôle EMRFS présentée sur cette page a été améliorée avec l'introduction d'Amazon S3 Access Grants dans Amazon EMR 6.15.0. Pour une solution de contrôle d'accès évolutive pour vos données dans Amazon S3, nous vous recommandons d'utiliser S3 Access Grants avec Amazon EMR.

Lorsqu'une application s'exécutant sur un cluster référence des données à l'aide du format `s3://mydata`, Amazon EMR utilise EMRFS pour effectuer la demande. Pour interagir avec Amazon S3, EMRFS assume les politiques d'autorisation qui sont attachées à votre profil d'instance Amazon EC2. Le même profil d'instance Amazon EC2 est utilisé quel que soit l'utilisateur ou le groupe qui exécute l'application ou l'emplacement des données dans Amazon S3.

Si vous avez un cluster 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 assumer un rôle de service différent pour les instances EC2 de cluster en fonction de l'utilisateur ou du groupe à l'origine de la demande, ou en fonction de l'emplacement des données dans Amazon S3. Chaque rôle IAM pour EMRFS peut avoir des autorisations différentes pour l'accès aux données dans Amazon S3. Pour plus d'informations sur le rôle de service pour les instances EC2 de cluster, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).

L'utilisation de rôles IAM personnalisés pour EMRFS est prise en charge dans les versions 5.10.0 et ultérieures d'Amazon EMR. 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.

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 à la place 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 effectuée via EMRFS ne correspond à aucun identifiant, EMRFS revient à l'utilisation 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 Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).

Configuration des rôles

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, consultez Comment fonctionnent les rôles pour les instances EC2 ? dans le Guide de l'utilisateur IAM. Lorsque vous créez des stratégies d'autorisations, nous vous recommandons de commencer par la stratégie gérée attachée au rôle Amazon EMR par défaut pour EC2 et de modifier cette stratégie en fonction de vos besoins. Le nom du rôle par défaut est `EMR_EC2_DefaultRole` et la stratégie gérée pour modifier la valeur par défaut est `AmazonElasticMapReduceforEC2Role`. Pour plus d'informations, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).

Mise à jour des politiques d'approbation pour prendre en charge les autorisations de rôle

Chaque rôle qu'EMRFS utilise doit disposer d'une stratégie d'approbation qui permet au rôle Amazon EMR du cluster pour EC2 de l'assumer. De même, le rôle Amazon 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 Amazon 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.

Spécification d'un rôle en tant qu'utilisateur de la clé

Si un rôle permet d'accéder à un emplacement dans Amazon S3 qui est chiffré à l'aide d'une AWS KMS key, assurez-vous que le rôle est spécifié en tant qu'utilisateur clé. Cela donne au rôle l'autorisation d'utiliser la clé KMS. Pour plus d'informations, consultez Politiques de clé dans AWS KMS dans le Guide du développeur AWS Key Management Service .

Définition d'une configuration de sécurité avec des rôles IAM pour EMRFS

> **Important**
>
> Si aucun des rôles IAM pour EMRFS que vous spécifiez ne s'applique, EMRFS bascule vers le rôle Amazon 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 plus d'informations, consultez Personnaliser les rôles IAM et Spécifiez les rôles IAM personnalisés lors de la création d'un cluster.

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 Configurations de sécurité, 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 (Utilisateurs, Groupes ou Préfixes S3) dans la liste et entrez 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 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.
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.

Pour spécifier les rôles IAM pour les demandes EMRFS adressées à Amazon S3 à l'aide du 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.

   L'exemple de commande ci-dessous crée une configuration de sécurité nommée `EMRFS_Roles_Security_Configuration`. Elle est basée sur une structure JSON dans le fichier `MyEmrfsSecConfig.json`, qui est enregistré dans le répertoire où la commande est exécutée.

   ```
   aws emr create-security-configuration --name EMRFS_Roles_Security_Configuration --
   security-configuration file://MyEmrFsSecConfig.json.
   ```

   Utilisez les instructions suivantes pour la structure du fichier `MyEmrFsSecConfig.json`. 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.

Vous trouverez ci-dessous un exemple d'extrait JSON permettant de spécifier des rôles IAM personnalisés pour EMRFS dans une configuration de sécurité. Il montre les mappages de rôles pour les trois types d'identifiants différents, suivis d'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 des mappages de rôles.

Paramètre	Description
`"RoleMappings":`	Obligatoire. Contient une ou plusieurs définitions de mappage de rôles. Les mappages de rôles sont évalués dans l'ordre d'apparition du haut vers le bas. Si un mappage de rôle est considéré comme vrai pour un appel de données EMRFS dans Amazon S3, aucun autre mappage de rôle n'est évalué et EMRFS utilise le rôle IAM spécifié pour la demande. Les mappages de rôles sont constitués des 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 envoyée à Amazon S3 correspond à l'une des `Identifiers` spécifiées.

Paramètre	Description
`"IdentifierType":`	Les valeurs suivantes sont possibles : • `"User"` indique que les identifiants sont ceux d'un ou de plusieurs utilisateurs Hadoop, qui peuvent être des utilisate urs de comptes Linux ou des utilisate urs principaux de Kerberos. Lorsque la demande EMRFS provient de l'utilisateur ou des utilisateurs spécifiés, le rôle IAM est assumé. • `"Prefix"` indique que l'identifiant est un emplacement Amazon S3. Le rôle IAM est assumé pour les appels vers le ou les emplacements dotés des préfixes spécifiés. Par exemple, le préfixe `s3://mybucket/` correspond à `s3://mybucket/mydir` et `s3://mybucket/yetanotherdir` . • `"Group"` indique que les identifiants sont un ou plusieurs groupes Hadoop. Le rôle IAM est assumé si la demande provient d'un utilisateur appartenant à un ou plusieurs groupes spécifiés.
`"Identifiers":`	Spécifie un ou plusieurs identifiants du type d'identifiant approprié. Séparez les identifia nts multiples 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 Amazon

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 continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name MyEmrFsS3RolesCluster \
--release-label emr-7.1.0 --ec2-attributes
 InstanceProfile=EC2_Role_EMR_Restrict_S3,KeyName=MyKey \
--instance-type m5.xlarge --instance-count 3 \
--security-configuration EMRFS_Roles_Security_Configuration
```

Utilisation de politiques basées sur les ressources pour l'accès d'Amazon EMR au catalogue de données AWS Glue

Si vous utilisez AWS Glue conjointement avec Hive, Spark ou Presto dans Amazon EMR, AWS Glue prend en charge les politiques basées sur les ressources afin de contrôler l'accès aux ressources du catalogue de données. Ces ressources comprennent les bases de données, les tables, les connexions et les fonctions définies par l'utilisateur. Pour plus d'informations, consultez Politiques de ressources AWS Glue dans le Guide du développeur AWS Glue.

Lorsque vous utilisez des politiques basées sur les ressources pour limiter l'accès à AWS Glue depuis Amazon EMR, le principal que vous spécifiez dans la politique d'autorisation doit être l'ARN du rôle associé au profil d'instance EC2 spécifié lors de la création d'un cluster. Par exemple, pour une politique basée sur les ressources attachée à un catalogue, vous pouvez spécifier le rôle ARN pour le rôle de service par défaut pour les instances EC2 du cluster, *EMR_EC2_ en DefaultRole tant que tel, en utilisant le format illustré* dans l'exemple `Principal` suivant :

```
arn:aws:iam::acct-id:role/EMR_EC2_DefaultRole
```

L'*acct-id* peut être différent de l'identifiant du compte AWS Glue. Cela permet d'accéder aux clusters EMR à partir de comptes différents. Vous pouvez spécifier plusieurs principaux, chacun provenant d'un compte différent.

Utilisation des rôles IAM avec des applications qui appellent directement les services AWS

Les applications exécutées sur les instances EC2 d'un cluster peuvent utiliser le profil d'instance EC2 pour obtenir des informations d'identification de sécurité temporaires lorsqu'elles appellent AWS des services.

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 uniquement sur l'architecture Hadoop et n'appelle directement aucun service AWS, elle devrait fonctionner avec les rôles IAM sans aucune modification.

Si votre application appelle AWS directement des services, 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 AWS ressources avec des rôles IAM à l'aide d'un SDK

- Les rubriques suivantes montrent comment utiliser plusieurs AWS SDK pour accéder à des 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.
  - Utilisation des rôles IAM pour les instances Amazon EC2 avec le kit SDK pour Java dans le Guide du développeur AWS SDK for Java .
  - Utilisation des rôles IAM pour les instances Amazon EC2 avec le kit SDK pour .NET dans le Guide du développeur AWS SDK for .NET .
  - Utilisation des rôles IAM pour les instances Amazon EC2 avec le kit SDK pour PHP dans le Guide du développeur AWS SDK for PHP .
  - Utilisation des rôles IAM pour les instances Amazon EC2 avec le kit SDK pour Ruby dans le Guide du développeur AWS SDK for Ruby .

Pour obtenir des informations d'identification temporaires à partir de métadonnées d'instance EC2

- Appelez l'URL suivante depuis une instance EC2 exécutée 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 utilisant des rôles IAM, consultez Accorder l'accès AWS aux ressources aux applications qui s'exécutent sur des instances Amazon EC2.

Pour plus d'informations sur les informations d'identification temporaires, consultez Utilisation des informations d'identification temporaires dans le guide Utilisation des informations d'identification temporaires.

Permettre aux utilisateurs et aux groupes de créer et de modifier des rôles

Les principaux 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 la Référence API IAM.

- `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.

Pour en savoir plus sur l'attribution d'autorisations à un utilisateur, consultez Modification des autorisations d'un utilisateur dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur une identité pour Amazon EMR

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources Amazon EMR. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques 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 politiques aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, veuillez consulter Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

- Bonnes pratiques en matière de politiques pour Amazon EMR
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Politiques gérées par Amazon EMR
- Politiques IAM pour l'accès basé sur des balises aux clusters et aux bloc-notes EMR
- Refuser l' ModifyInstanceGroup action
- Résolution de problèmes pour identité et accès Amazon EMR

Bonnes pratiques en matière de politiques pour Amazon EMR

Les politiques 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 AWS compte. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez à utiliser les politiques AWS gérées — Pour commencer à utiliser Amazon EMR rapidement, utilisez des politiques AWS gérées pour donner à vos employés les autorisations dont ils ont besoin. Ces politiques sont déjà disponibles dans votre compte et sont gérées et mises à jour par AWS. Pour plus d'informations, consultez Commencer à utiliser les autorisations avec des politiques AWS gérées dans le guide de l'utilisateur IAM etPolitiques gérées par Amazon EMR.

- Accorder le privilège le plus faible : Lorsque vous créez des politiques personnalisées, accordez uniquement les autorisations requises pour exécuter une seule tâche. Commencez avec un ensemble d'autorisations minimum 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 moindre privilège possible dans le Guide de l'utilisateur IAM.
- Activer la MFA pour les opérations sensibles – Pour plus de sécurité, obligez les utilisateurs à utiliser l'authentification multifactorielle (MFA) pour accéder à des ressources ou à des opérations d'API sensibles. Pour plus d'informations, consultez Utilisation de l'authentification multifacteur (MFA) dans AWS dans le Guide de l'utilisateur IAM.
- Utiliser des conditions de politique pour une plus grande sécurité : tant que cela reste pratique pour vous, définissez les conditions dans lesquelles vos politiques 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 de plus amples informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ViewOwnUserInfo",
         "Effect":"Allow",
         "Action":[
            "iam:GetUser",
            "iam:GetUserPolicy",
            "iam:ListAttachedUserPolicies",
            "iam:ListGroupsForUser",
            "iam:ListUserPolicies"
         ],
```

```
         "Resource":[
            "arn:aws:iam::*:user/${aws:username}"
         ]
      },
      {
         "Sid":"NavigateInConsole",
         "Effect":"Allow",
         "Action":[
            "iam:GetGroupPolicy",
            "iam:GetPolicy",
            "iam:GetPolicyVersion",
            "iam:ListAttachedGroupPolicies",
            "iam:ListGroupPolicies",
            "iam:ListGroups",
            "iam:ListPolicies",
            "iam:ListPolicyVersions",
            "iam:ListUsers"
         ],
         "Resource":"*"
      }
   ]
}
```

Politiques gérées par Amazon EMR

Le moyen le plus simple d'accorder un accès complet ou un accès en lecture seule aux actions requises d'Amazon EMR est d'utiliser les politiques 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.

Amazon EMR va supprimer les politiques gérées existantes (politiques v1) au profit de nouvelles politiques gérées (politiques v2). Les nouvelles politiques gérées ont été réduites afin de s'aligner sur les AWS meilleures pratiques. Une fois que les politiques gérées v1 existantes auront été supprimées, vous ne pourrez plus les associer à de nouveaux rôles ou utilisateurs IAM. Les rôles et utilisateurs existants qui utilisent des politiques obsolètes peuvent continuer à les utiliser. Les politiques gérées v2 restreignent l'accès au moyen de balises. Elles n'autorisent que les actions Amazon EMR spécifiées et nécessitent des ressources de cluster étiquetées avec une clé spécifique à EMR. Nous vous recommandons de lire attentivement la documentation avant d'utiliser les nouvelles politiques v2.

Les politiques v1 seront marquées comme étant obsolètes avec une icône d'avertissement à côté d'elles dans la liste Politiques de la console IAM. Les politiques obsolètes auront les caractéristiques suivantes :

- Elles continueront à fonctionner pour tous les utilisateurs, groupes et rôles actuellement attachés. Aucun élément ne cesse de fonctionner.
- Elles ne peuvent pas être attachées à de nouveaux utilisateurs, groupes ou rôles. Si vous détachez l'une des politiques d'une entité actuelle, vous ne pouvez pas la rattacher.
- Une fois que vous avez détaché une politique v1 de toutes les entités actuelles, la politique n'est plus visible et ne peut plus être utilisée.

Le tableau suivant récapitule les modifications entre les politiques actuelles (v1) et les politiques v2.

Amazon EMR a géré les modifications de politique

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
Rôle du service EMR par défaut et politique gérée associée	Nom du rôle : EMR_DefaultRole Politique V1 (à déconseiller) : AmazonElasticMapReducerôle (rôle de service EMR) Nom de la politique V2 (limitée) : **AmazonEMRServicePolicy_v2**	Permet à Amazon EMR d'appeler d'autres AWS services en votre nom lors de la mise en service des ressources et de l'exécution d'actions au niveau des services. Ce rôle est obligatoire pour tous les clusters.	La politique ajoute la nouvelle autorisation `"ec2:DescribeInstanceTypeOfferings"` . Cette opération d'API renvoie une liste de types d'instances pris en charge par une liste de zones de disponibilité données.
Politique gérée par IAM pour un accès complet à Amazon EMR par utilisate	Nom de la politique V2 (limitée) : **AmazonEMR**	Accorde aux utilisateurs des autorisations complètes pour les actions EMR.	La politique ajoute une condition préalable selon laquelle les utilisate

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
ur, rôle ou groupe attaché	**ServicePolicy_v2**	Inclut iam : PassRole autorisations pour les ressources.	urs doivent ajouter des balises d'utilisateur aux ressources avant de pouvoir utiliser cette politique. veuillez consulter Balisage des ressources pour l'utilisation des politiques gérées. iam : PassRole action nécessite que la PassedToService condition iam : soit définie sur le service spécifié. L'accès à Amazon EC2, Amazon S3 et à d'autres services n'est pas autorisé par défaut. Consultez Politique IAM gérée pour un accès complet (politique gérée par défaut v2).

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
Politique IAM pour l'accès en lecture seule par l'utilisateur, le rôle ou le groupe attaché.	Politique V1 (qui sera obsolète) : **AmazonElasticMapReduceReadOnlyAccess** Nom de la politique V2 (limitée) : **AmazonEMRReadOnlyAccessPolicy_v2**	Accorde aux utilisateurs des autorisations en lecture seule pour les actions Amazon EMR.	Les autorisations ne permettent que les actions en lecture seule spécifiées d'elasticmapreduce. L'accès à Amazon S3 n'est pas autorisé par défaut. Consultez Politique IAM gérée pour l'accès en lecture seule (politique gérée par défaut v2).

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
Rôle du service EMR par défaut et politique gérée associée	Nom du rôle : EMR_DefaultRole Politique V1 (à déconseiller) : AmazonElasticMapReducerôle (rôle de service EMR) Nom de la politique V2 (limitée) : **AmazonEMRServicePolicy_v2**	Permet à Amazon EMR d'appeler d'autres AWS services en votre nom lors de la mise en service des ressources et de l'exécution d'actions au niveau des services. Ce rôle est obligatoire pour tous les clusters.	Le rôle de service v2 et la politique par défaut v2 remplacent le rôle et la politique obsolètes. La politique ajoute une condition préalable selon laquelle les utilisateurs doivent ajouter des balises d'utilisateur aux ressources avant de pouvoir utiliser cette politique. veuillez consulter Balisage des ressources pour l'utilisation des politiques gérées. veuillez consulter Rôle de service pour Amazon EMR (rôle EMR).

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
Rôle de service pour les instances EC2 de cluster (profil d'instance EC2)	Politique V1 (à déconseiller) : DefaultRoleEMR_EC2_ (profil d'instance) Nom de la politique obsolète : EC2Role AmazonElastic MapReducefor	Permet aux applications exécutées sur un cluster EMR d'accéder à d'autres ressources AWS , telles qu'Amazon S3. Par exemple, si vous exécutez des tâches Apache Spark qui traitent des données provenant d'Amazon S3, la politique doit autoriser l'accès à ces ressources.	Le rôle et la politique par défaut sont tous deux sur le point d'être obsolètes. Il n'existe aucun rôle ou politique géré AWS par défaut de remplacement. Vous devez fournir une politique basée sur les ressources ou sur l'identité. Cela signifie que, par défaut, les applications s'exécutant sur un cluster EMR n'ont pas accès à Amazon S3 ou à d'autres ressources, à moins que vous ne les ajoutiez manuellement à la politique. veuillez consulter Rôle et stratégie gérée par défaut.

Type de stratégie	Noms des politiques	Objectif de la politique	Modifications apportées à la politique v2
Autres politiques relatives aux rôles de service EC2	Noms des politiques actuelles : AmazonElasticMapReduceforAutoScalingRole AmazonElasticMapReduceEditorsRole, AmazonEMR CleanupPolicy	Fournit les autorisations dont Amazon EMR a besoin pour accéder à d'autres AWS ressources et effectuer des actions en cas d'utilisation du dimensionnement automatique, de blocs-notes ou pour nettoyer les ressources EC2.	Aucun changement pour la version 2.

Fixation de l'objectif : PassRole

Les politiques gérées par défaut d'Amazon EMR avec autorisations complètes intègrent des configurations de sécurité `iam:PassRole`, notamment les suivantes :

- Les autorisations `iam:PassRole` uniquement pour des rôles Amazon EMR par défaut spécifiques.
- `iam:PassedToService`conditions qui vous permettent d'utiliser la politique uniquement avec AWS des services spécifiques, tels que `elasticmapreduce.amazonaws.com` et`ec2.amazonaws.com`.

Vous pouvez consulter la version JSON des politiques AmazonEMR FullAccess Policy_v2 et AmazonEMR ServicePolicy _v2 dans la console IAM. Nous vous recommandons de créer de nouveaux clusters avec les politiques gérées v2.

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 manière d'attacher des politiques à des utilisateurs (principaux), consultez Utilisation de politiques gérées à l'aide de la AWS Management Console dans le Guide de l'utilisateur IAM.

Balisage des ressources pour l'utilisation des politiques gérées

AmazonEMR ServicePolicy _v2 et AmazonEMR FullAccess Policy_v2 dépendent d'un accès limité aux ressources qu'Amazon EMR fournit ou utilise. La réduction de la portée est obtenue en limitant l'accès aux seules ressources associées à une balise utilisateur prédéfinie. Lorsque vous utilisez l'une de ces deux politiques, vous devez transmettre la balise utilisateur prédéfinie `for-use-with-amazon-emr-managed-policies = true` lorsque vous provisionnez le cluster. Amazon EMR propagera alors automatiquement ces balises. Vous devez également ajouter une balise utilisateur aux ressources énumérées dans la section suivante. Si vous utilisez la console Amazon EMR pour lancer votre cluster, consultez Considérations relatives à l'utilisation de la console Amazon EMR pour lancer des clusters avec des politiques gérées v2.

Pour utiliser des politiques gérées, transmettez la balise utilisateur `for-use-with-amazon-emr-managed-policies = true` lorsque vous provisionnez un cluster à l'aide de la CLI, du kit SDK ou d'une autre méthode.

Lorsque vous transmettez la balise, Amazon EMR propage la balise vers le sous-réseau privé ENI, l'instance EC2 et les volumes EBS qu'il crée. Amazon EMR balise également automatiquement les groupes de sécurité qu'il crée. Toutefois, si vous voulez qu'Amazon EMR soit lancé avec un certain groupe de sécurité, vous devez le baliser. Pour les ressources qui ne sont pas créées par Amazon EMR, vous devez ajouter des balises à ces ressources. Par exemple, vous devez baliser les sous-réseaux Amazon EC2, les groupes de sécurité EC2 (s'ils ne sont pas créés par Amazon EMR) et les VPC (si vous voulez qu'Amazon EMR crée des groupes de sécurité). Pour lancer des clusters avec des politiques gérées v2 dans des VPC, vous devez baliser ces VPC avec la balise utilisateur prédéfinie. Consultez Considérations relatives à l'utilisation de la console Amazon EMR pour lancer des clusters avec des politiques gérées v2.

Balisage propagé spécifié par l'utilisateur

Amazon EMR balise les ressources qu'il crée à l'aide des balises Amazon EMR que vous spécifiez lors de la création d'un cluster. Amazon EMR applique des balises aux ressources qu'il crée pendant la durée de vie du cluster.

Amazon EMR propage les balises utilisateur pour les ressources suivantes :

- Sous-réseau privé ENI (interfaces réseau élastiques d'accès aux services)
- Instances EC2
- Volumes EBS

- Modèle de lancement EC2

Groupes de sécurité balisés automatiquement

Amazon EMR balise les groupes de sécurité EC2 qu'il crée avec la balise requise pour les politiques gérées v2 pour Amazon EMR, `for-use-with-amazon-emr-managed-policies`, quelles que soient les balises que vous spécifiez dans la commande de création de cluster. Pour un groupe de sécurité créé avant l'introduction des politiques gérées v2, Amazon EMR ne balise pas automatiquement le groupe de sécurité. Si vous voulez utiliser des politiques gérées v2 avec les groupes de sécurité par défaut qui existent déjà dans le compte, vous devez baliser les groupes de sécurité manuellement avec `for-use-with-amazon-emr-managed-policies = true`.

Ressources de cluster balisées manuellement

Vous devez baliser manuellement certaines ressources du cluster afin que les rôles par défaut d'Amazon EMR puissent y accéder.

- Vous devez baliser manuellement les groupes de sécurité EC2 et les sous-réseaux EC2 avec la balise de politique gérée Amazon EMR `for-use-with-amazon-emr-managed-policies`.
- Vous devez baliser manuellement un VPC si vous voulez qu'Amazon EMR crée des groupes de sécurité par défaut. EMR essaiera de créer un groupe de sécurité avec la balise spécifique si le groupe de sécurité par défaut n'existe pas déjà.

Amazon EMR balise automatiquement les ressources suivantes :

- Groupes de sécurité EC2 créés par EMR

Vous devez baliser manuellement les ressources suivantes :

- Sous-réseau EC2
- Groupes de sécurité EC2

Vous pouvez, en option, baliser manuellement les ressources suivantes :

- VPC : uniquement lorsque vous voulez qu'Amazon EMR crée des groupes de sécurité

Considérations relatives à l'utilisation de la console Amazon EMR pour lancer des clusters avec des politiques gérées v2

Vous pouvez provisionner des clusters avec des politiques gérées v2 à l'aide de la console Amazon EMR. Voici quelques points à prendre en compte lorsque vous utilisez la console pour lancer des clusters Amazon EMR.

> **Note**
>
> Nous avons repensé la console Amazon EMR. La fonctionnalité de balisage automatique n'est pas encore disponible dans la nouvelle console, et la nouvelle console ne vous indique pas non plus quelles ressources (VPC/sous-réseaux) doivent être balisées. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

- Il n'est pas nécessaire de transmettre la balise prédéfinie. Amazon EMR ajoute automatiquement la balise et la propage aux composants appropriés.
- Pour les composants qui doivent être balisés manuellement, l'ancienne console Amazon EMR essaie de les baliser automatiquement si vous disposez des autorisations requises pour baliser les ressources. Si vous n'êtes pas autorisé à baliser les ressources ou si vous voulez utiliser la nouvelle console, demandez à votre administrateur de baliser ces ressources.
- Vous ne pouvez pas lancer de clusters avec des politiques gérées v2 si toutes les conditions préalables ne sont pas remplies.
- L'ancienne console Amazon EMR vous indique quelles ressources (VPC/sous-réseaux) doivent être balisées.

Politique gérée par IAM pour un accès complet (politique par défaut gérée v2)

Les politiques gérées par défaut EMR limitées à la v2 accordent des privilèges d'accès spécifiques aux utilisateurs. Elles nécessitent une balise de ressource Amazon EMR prédéfinie et des clés de condition `iam:PassRole` pour les ressources utilisées par Amazon EMR, telles que `Subnet` et `SecurityGroup` que vous utilisez pour lancer votre cluster.

Pour accorder les actions requises limitées à Amazon EMR, attachez la politique gérée par `AmazonEMRFullAccessPolicy_v2`. Cette politique gérée par défaut mise à jour remplace la politique gérée `AmazonElasticMapReduceFullAccess`.

`AmazonEMRFullAccessPolicy_v2` dépend de l'accès limité aux ressources qu'Amazon EMR fournit ou utilise. Lorsque vous utilisez cette politique, vous devez transmettre la balise utilisateur `for-use-with-amazon-emr-managed-policies = true` lors du provisionnement du cluster. Amazon EMR propagera automatiquement la balise. Vous pouvez également avoir besoin d'ajouter manuellement une balise utilisateur à des types de ressources spécifiques, tels que les groupes de sécurité EC2 qui n'ont pas été créés par Amazon EMR. Pour plus d'informations, consultez Balisage des ressources pour l'utilisation des politiques gérées.

La politique `AmazonEMRFullAccessPolicy_v2` sécurise les ressources en procédant comme suit :

- Nécessite que les ressources soient balisées avec la balise prédéfinie des politiques gérées par Amazon EMR `for-use-with-amazon-emr-managed-policies` pour la création de clusters et l'accès à Amazon EMR.
- Limite l'action `iam:PassRole` à des rôles par défaut spécifiques et l'accès `iam:PassedToService` à des services spécifiques.
- Ne donne plus accès à Amazon EC2, Amazon S3 et à d'autres services par défaut.

Voici le contenu de cette politique.

> **Note**
>
> Vous pouvez également utiliser le lien de la console `AmazonEMRFullAccessPolicy_v2` pour afficher la politique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RunJobFlowExplicitlyWithEMRManagedTag",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
                }
```

```
            }
        },
        {
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:AddInstanceFleet",
                "elasticmapreduce:AddInstanceGroups",
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:AddTags",
                "elasticmapreduce:CancelSteps",
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:CreateSecurityConfiguration",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:DeleteSecurityConfiguration",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ModifyCluster",
                "elasticmapreduce:ModifyInstanceFleet",
                "elasticmapreduce:ModifyInstanceGroups",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:PutAutoScalingPolicy",
                "elasticmapreduce:PutBlockPublicAccessConfiguration",
                "elasticmapreduce:PutManagedScalingPolicy",
                "elasticmapreduce:RemoveAutoScalingPolicy",
                "elasticmapreduce:RemoveManagedScalingPolicy",
                "elasticmapreduce:RemoveTags",
                "elasticmapreduce:SetTerminationProtection",
```

```
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewMetricsInEMRConsole",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/EMR_DefaultRole",
                "arn:aws:iam::*:role/EMR_DefaultRole_V2"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForAutoScaling",
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "application-autoscaling.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "ElasticMapReduceServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
elasticmapreduce.amazonaws.com*/AWSServiceRoleForEMRCleanup*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "elasticmapreduce.amazonaws.com",
                        "elasticmapreduce.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Sid": "ConsoleUIActions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNatGateways",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "s3:ListAllMyBuckets",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

Politique gérée par IAM pour un accès complet (sur le point de devenir obsolète)

Les politiques gérées `AmazonElasticMapReduceFullAccess` et `AmazonEMRFullAccessPolicy_v2` AWS Identity and Access Management (IAM) accordent toutes les actions requises pour Amazon EMR et d'autres services.

> **Important**
>
> La politique gérée `AmazonElasticMapReduceFullAccess` est sur le point de devenir obsolète et son utilisation avec Amazon EMR n'est plus recommandée. Utilisez à la place `AmazonEMRFullAccessPolicy_v2`. Lorsque le service IAM rendra la politique v1 obsolète, vous ne pourrez plus l'attacher à un rôle. Toutefois, vous pouvez associer un rôle existant à un cluster même si ce rôle utilise la politique obsolète.

Les politiques gérées par défaut d'Amazon EMR avec autorisations complètes intègrent des configurations de sécurité `iam:PassRole`, notamment les suivantes :

- Les autorisations `iam:PassRole` uniquement pour des rôles Amazon EMR par défaut spécifiques.
- `iam:PassedToService`conditions qui vous permettent d'utiliser la politique uniquement avec AWS des services spécifiques, tels que `elasticmapreduce.amazonaws.com` et`ec2.amazonaws.com`.

Vous pouvez consulter la version JSON des politiques AmazonEMR FullAccess Policy_v2 et AmazonEMR ServicePolicy _v2 dans la console IAM. Nous vous recommandons de créer de nouveaux clusters avec les politiques gérées v2.

Vous pouvez consulter le contenu de la politique obsolète v1 dans le AWS Management Console fichier at. `AmazonElasticMapReduceFullAccess` L'action `ec2:TerminateInstances` prévue dans la politique autorise un utilisateur ou un rôle à résilier l'une des instances Amazon EC2 associées au compte IAM. Cela inclut les instances qui ne font pas partie d'un cluster EMR.

Politique IAM gérée pour l'accès en lecture seule (politique gérée par défaut v2).

Pour accorder des privilèges de lecture seule à Amazon EMR, joignez la politique gérée ReadOnlyAccessPolicyAmazonEMR _v2. Cette politique gérée par défaut remplace la politique gérée `AmazonElasticMapReduceReadOnlyAccess`.
Le contenu de cette déclaration de politique est repris dans l'extrait suivant. Par

rapport à la politique `AmazonElasticMapReduceReadOnlyAccess`, la politique `AmazonEMRReadOnlyAccessPolicy_v2` n'utilise pas de caractères génériques pour l'élément `elasticmapreduce`. Au lieu de cela, la politique v2 par défaut définit le champ d'application des actions `elasticmapreduce` autorisées.

> **Note**
>
> Vous pouvez également utiliser le AWS Management Console lien `AmazonEMRReadOnlyAccessPolicy_v2`pour consulter la politique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewMetricsInEMRConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        }
    ]
}
```

Politique IAM gérée pour l'accès en lecture seule (sur le point de devenir obsolète)

La politique gérée `AmazonElasticMapReduceReadOnlyAccess` est sur le point de devenir obsolète. Vous ne pouvez pas attacher cette politique lors du lancement de nouveaux clusters. `AmazonElasticMapReduceReadOnlyAccess` a été remplacé par `AmazonEMRReadOnlyAccessPolicy_v2` en tant que politique gérée par défaut d'Amazon EMR. Le contenu de cette déclaration de politique est repris dans l'extrait suivant. 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**
>
> Vous pouvez également utiliser le AWS Management Console pour consulter la politique.

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

AWS politique gérée : EMR EMRWAL DescribeCluster PolicyFor

Vous ne pouvez pas joindre de `EMRDescribeClusterPolicyForEMRWAL` à vos entités IAM. Cette politique est associée à un rôle lié à un service qui permet à Amazon EMR d'effectuer des actions en votre nom. Pour plus d'informations sur ce rôle lié à un service, consultez. Utilisation de rôles liés à un service pour la journalisation anticipée

Cette politique accorde des autorisations en lecture seule qui permettent au service WAL pour Amazon EMR de rechercher et de renvoyer l'état d'un cluster. Pour plus d'informations sur Amazon EMR WAL, consultez la section Write ahead logs (WAL) pour Amazon EMR.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes :

- `emr`— Permet aux principaux de décrire l'état du cluster à partir d'Amazon EMR. Cela est nécessaire pour qu'Amazon EMR puisse confirmer la fermeture d'un cluster, puis, au bout de trente jours, nettoyer tous les journaux WAL laissés par le cluster.

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

AWS politiques gérées pour Amazon EMR

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation

courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez la section Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Mises à jour des politiques gérées par Amazon EMR AWS

Consultez les informations relatives aux mises à jour des politiques AWS gérées pour Amazon EMR depuis que ce service a commencé à suivre ces modifications.

Modification	Description	Date
EMRDescribeClusterPolicyForEMRWAL : nouvelle politique	Ajout d'une nouvelle politique afin qu'Amazon EMR puisse déterminer l'état du cluster pour le nettoyage du WAL trente jours après la fin du cluster.	10 août 2023
AmazonEMRFullAccessPolicy_v2 et **AmazonEMRReadOnlyAccessPolicy_v2** : mise à jour d'une stratégie existante	Ajout de `elasticmapreduce:DescribeReleaseLabel` et de `elasticmapreduce:GetAutoTerminationPolicy` .	21 avril 2022

Modification	Description	Date
AmazonEMRFullAccessPolicy_v2 – Mise à jour d'une politique existante	Ajout de `ec2:DescribeImages` pour Utilisation d'une image AMI personnalisée.	15 février 2022
Politiques gérées par Amazon EMR	Mis à jour pour clarifier l'utilisation de balises utilisateur prédéfinies. Ajout d'une section sur l'utilisation de la AWS console pour lancer des clusters avec des politiques gérées dans la version 2.	29 septembre 2021
AmazonEMRFullAccessPolicy_v2 – Mise à jour d'une politique existante	Modification des actions `PassRoleForAutoScaling` et `PassRoleForEC2` pour utiliser l'opérateur de condition `StringLike` pour correspondre à `"iam:PassedToService":"application-autoscaling.amazonaws.com*"` et `"iam:PassedToService":"ec2.amazonaws.com*"`, respectivement.	20 mai 2021

Modification	Description	Date
AmazonEMRFullAccessPolicy_v2 – Mise à jour d'une politique existante	L'action non valide `s3:ListBuckets` a été supprimée et remplacée par l'action `s3:ListAllMyBuckets` . La création de rôles liés à un service (SLR) a été mise à jour pour être explicitement limitée au seul SLR dont dispose Amazon EMR avec des principes de service explicites. Les SLR qui peuvent être créés sont exactement les mêmes qu'avant cette modification.	23 mars 2021

Modification	Description	Date
AmazonEMRFullAccessPolicy_v2 : nouvelle politique	Amazon EMR a ajouté de nouvelles autorisations pour limiter l'accès aux ressources et pour ajouter une condition préalable selon laquelle les utilisateurs doivent ajouter une balise utilisateur prédéfinie aux ressources avant de pouvoir utiliser les politiques gérées par Amazon EMR. L'action `iam:PassRole` nécessite que la condition `iam:PassedToService` soit définie sur le service spécifié. L'accès à Amazon EC2, Amazon S3 et à d'autres services n'est pas autorisé par défaut.	11 mars 2021
AmazonEMRServicePolicy_v2 : nouvelle politique	Ajoute une condition préalable selon laquelle les utilisateurs doivent ajouter des balises utilisateur aux ressources avant de pouvoir utiliser cette politique.	11 mars 2021
AmazonEMRReadOnlyAccessPolicy_v2 : nouvelle politique	Les autorisations ne permettent que les actions en lecture seule spécifiées d'elasticmapreduce. L'accès à Amazon S3 n'est pas autorisé par défaut.	11 mars 2021

Modification	Description	Date
Amazon EMR a commencé à assurer le suivi des modifications	Amazon EMR a commencé à suivre les modifications apportées à ses politiques AWS gérées.	11 mars 2021

Politiques 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.

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 politique 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 le test des politiques IAM, consultez le Guide de l'utilisateur IAM.

> **Important**
>
> Le refus explicite d'autoriser l'attribution de balises doit être pris en considération. Cela empêche les utilisateurs de baliser une ressource et de s'accorder ainsi des autorisations que vous n'aviez pas l'intention d'accorder. Si vous ne refusez pas les actions de balisage pour une ressource, un utilisateur peut modifier les balises et contourner l'intention des politiques basées sur les balises.

Exemple de déclarations de stratégie basées sur les identités pour les clusters

Les exemples suivants illustrent les stratégies d'autorisation basées sur une identité qui sont utilisées pour contrôler les actions autorisées avec les clusters EMR.

> **Important**
>
> L'action `ModifyInstanceGroup` dans Amazon EMR ne nécessite pas que vous spécifiiez un ID de cluster. Pour cette raison, le refus de cette action sur la base des balises de cluster nécessite une attention supplémentaire. Pour plus d'informations, consultez Refuser l' ModifyInstanceGroup action.

Rubriques

- Actions autorisées uniquement sur des clusters avec des valeurs de balises spécifiques
- Exiger le balisage du cluster lors de la création d'un cluster
- Actions autorisées sur des clusters avec une balise spécifique, quelle que soit la valeur de la balise

Actions autorisées uniquement sur des clusters avec des valeurs de balises spécifiques

Les exemples suivants 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 suivant de la stratégie, l'opérateur de condition `StringEquals` essaie de faire correspondre *`dev`* avec la valeur de la balise *`department`*. Si la balise *`department`* n'a pas été ajoutée 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"]
              }
            }
```

Exiger le balisage du cluster lors de la création d'un cluster

Comme dans l'exemple précédent, l'exemple de politique suivant recherche la même balise correspondante : la valeur *dev* pour la balise *department*. Mais dans cet exemple, la clé de condition `RequestTag` indique que la politique s'applique lors de la création de balises. Vous devez donc créer un cluster avec une balise correspondant à la valeur spécifiée.

Pour créer un cluster avec une balise, vous devez également être autorisé à effectuer l'action `elasticmapredue:AddTags`. Pour cette déclaration, la clé de condition `elasticmapreduce:ResourceTag` garantit que IAM n'accorde l'accès qu'aux ressources à balises ayant la valeur *dev* sur la balise *department*. L'élément `Resource` est utilisé pour limiter cette autorisation aux ressources cluster.

Pour les `PassRole` ressources, vous devez fournir l'identifiant ou l'alias du AWS compte, le nom du rôle de service dans l'`PassRoleForEMR`instruction et le nom du profil d'instance dans l'`PassRoleForEC2`instruction. Pour plus d'informations sur le format ARN IAM, consultez ARN IAM du Guide de l'utilisateur IAM.

Pour plus d'informations sur la correspondance des valeurs clés des balises, consultez `aws:RequestTag/tag-key` dans le Guide de l'utilisateur IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "RunJobFlowExplicitlyWithTag",
        "Effect": "Allow",
        "Action": [
          "elasticmapreduce:RunJobFlow"
        ],
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "aws:RequestTag/department": "dev"
          }
        }
      },
      {
        "Sid": "AddTagsForDevClusters",
        "Effect": "Allow",
        "Action": "elasticmapreduce:AddTags",
        "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
        "Condition": {
          "StringEquals": {
            "elasticmapreduce:ResourceTag/department": "dev"
          }
        }
      },
      {
        "Sid": "PassRoleForEMR",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
```

```
          }
        }
      },
      {
        "Sid": "PassRoleForEC2",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "ec2.amazonaws.com*"
          }
        }
      }
    ]
}
```

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 Opérateur de condition pour vérifier l'existence des clés de condition dans le Guide de l'utilisateur IAM. 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. Pour la `PassRole` ressource, vous devez fournir l'identifiant ou l'alias du AWS compte, ainsi que le nom du rôle de service. Pour plus d'informations sur le format ARN IAM, consultez ARN IAM du Guide de l'utilisateur IAM.

Pour plus d'informations sur la spécification de l'opérateur de condition null (« false »), consultez la section Opérateur de condition pour vérifier l'existence des clés de condition dans le Guide de l'utilisateur IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateClusterTagNullCondition",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/department": "false"
                }
            }
        },
        {
            "Sid": "AddTagsNullCondition",
            "Effect": "Allow",
            "Action": "elasticmapreduce:AddTags",
            "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
            "Condition": {
                "Null": {
                    "elasticmapreduce:ResourceTag/department": "false"
                }
            }
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
```

```
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
                }
            }
        }
    ]
}
```

Exemple de déclarations de stratégie basées sur l'identité pour les bloc-notes EMR

Les exemples de politiques IAM présentés dans cette section illustrent des scénarios courants d'utilisation de clés pour limiter les actions autorisées à l'aide de Blocs-notes EMR. Dans la mesure où aucune autre stratégie associée au principal (utilisateur) autorise les actions, les clés de contexte de condition limitent les actions autorisées comme indiqué.

Example – Autorise l'accès uniquement à Blocs-notes EMR créé par un utilisateur sur la base d'un balisage

L'exemple de déclaration de politique suivant, lorsqu'il est attaché à un rôle ou à un utilisateur, permet à l'utilisateur de travailler uniquement avec les blocs-notes qu'il a créés. Cette déclaration de stratégie utilise la balise par défaut appliquée lorsqu'un bloc-notes est créé.

Dans l'exemple, l'opérateur de condition `StringEquals` essaie de faire correspondre une variable représentant l'ID d'utilisateur actuel (`{aws:userId}`) avec la valeur de la balise `creatorUserID`. Si la balise `creatorUserID` n'a pas été ajoutée 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 nécessite que l'utilisateur crée le cluster avec une balise dont la chaîne de clé `dept` et une valeur sont définies 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 – Limiter la création de bloc-notes aux clusters balisés et exiger 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 – Limiter 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 – Limiter 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"
                    ]
                }
            }
        }
    ]
}
```

Refuser l' ModifyInstanceGroup action

L'action ModifyInstanceGroupes 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'instance. Pour cette raison, une politique de refus apparemment simple pour cette action basée sur l'ID de cluster ou une balise de cluster peut ne pas avoir l'effet escompté. Examinons l'exemple de politique 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 auquel cette politique est attachée effectue une action `ModifyInstanceGroup` et ne spécifie que l'ID du groupe d'instance, la politique ne s'applique pas. Comme l'action est autorisée sur toutes les autres ressources, l'action est réussie.

Une solution à ce problème consiste à joindre une déclaration de politique à l'identité qui utilise un NotResourceélément pour refuser toute `ModifyInstanceGroup` action émise sans identifiant de cluster. L'exemple de politique suivant ajoute une telle instruction de refus de sorte que toute demande `ModifyInstanceGroups` échoue à moins qu'un identifiant de cluster ne soit spécifié. Étant donné qu'une identité doit spécifier un ID de cluster avec l'action, les déclarations de refus basées sur l'ID de cluster sont donc efficaces.

```
{
 "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 se pose lorsque vous voulez refuser l'action `ModifyInstanceGroups` en fonction de la valeur associée à une balise de cluster. La solution est similaire. En plus d'une instruction de refus qui spécifie la valeur de la balise, vous pouvez ajouter une instruction de stratégie qui refuse l'action `ModifyInstanceGroup` si la balise que vous spécifiez n'est pas présente, quelle qu'en soit la valeur.

L'exemple suivant illustre une politique qui, lorsqu'elle est attachée à une identité, refuse à cette dernière l'action `ModifyInstanceGroups` pour tout cluster dont la balise `department` est définie sur `dev`. Cette instruction n'est efficace qu'en raison de l'instruction de refus qui utilise la condition `StringNotLike` pour refuser l'action à moins que la balise `department` ne soit présente.

```
{
    "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": "*"
        }
    ],
}
```

Résolution de problèmes pour identité et accès Amazon EMR

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous utilisez Amazon EMR et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Amazon EMR
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon EMR

Je ne suis pas autorisé à effectuer une action dans Amazon EMR

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d’utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit quand l'utilisateur `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `EMR:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 EMR:GetWidget on resource: my-example-widget
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-widget`* à l'aide de l'action `EMR:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter l'action `iam:PassRole`, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon EMR.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon EMR. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon EMR

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques

basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Amazon EMR est compatible avec ces fonctionnalités, veuillez consulter Fonctionnement d'Amazon EMR avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Utilisation d'Amazon S3 Access Grants avec Amazon EMR

Présentation de S3 Access Grants pour Amazon EMR

Disponible à partir de la version 6.15.0 d'Amazon EMR, Amazon S3 Access Grants fournit une solution de contrôle d'accès évolutive que vous pouvez utiliser pour augmenter l'accès à vos données Amazon S3 depuis Amazon EMR. Si vous disposez d'une configuration d'autorisations complexe ou importante, vous pouvez utiliser Access Grants pour mettre à l'échelle les autorisations de données S3 pour les utilisateurs, les rôles et les applications de votre cluster.

Utilisez S3 Access Grants pour augmenter l'accès aux données Amazon S3 au-delà des autorisations accordées par le rôle d'exécution ou les rôles IAM associés aux identités ayant accès à votre cluster EMR. Pour plus d'informations, voir la rubrique Gestion des accès avec S3 Access Grants du Guide de l'utilisateur Amazon S3.

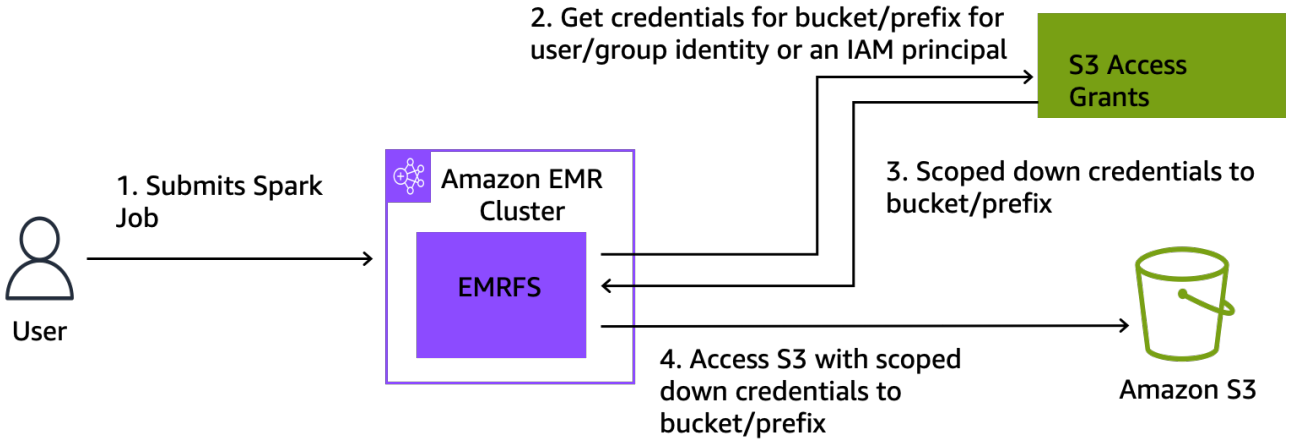
Pour savoir comment utiliser S3 Access Grants avec d'autres déploiements Amazon EMR, consultez la documentation suivante :

- Utilisation de S3 Access Grants avec Amazon EMR sur EKS
- Utilisation de S3 Access Grants avec Amazon EMR sans serveur

Fonctionnement d'Amazon EMR avec S3 Access Grants

La version 6.15.0 et les versions ultérieures d'Amazon EMR fournissent une intégration native avec S3 Access Grants. Vous pouvez activer S3 Access Grants sur Amazon EMR et exécuter des tâches Spark. Lorsqu'une tâche Spark demande à accéder aux données S3, Amazon S3 fournit des informations d'identification temporaires limitées au compartiment, au préfixe ou à l'objet concerné.

Voici un aperçu général de la manière dont Amazon EMR accède aux données protégées par S3 Access Grants.



1. Un utilisateur soumet une tâche Amazon EMR Spark qui utilise des données stockées dans Amazon S3.
2. Amazon EMR demande à S3 Access Grants l'autorisation d'accéder au compartiment, au préfixe ou à l'objet concerné au nom de cet utilisateur.
3. Amazon S3 renvoie des informations d'identification temporaires sous la forme d'un jeton AWS Security Token Service (STS) pour l'utilisateur. Le jeton permet uniquement d'accéder au compartiment, au préfixe ou à l'objet S3 concerné.
4. Amazon EMR utilise le jeton STS pour récupérer les données de S3.
5. Amazon EMR reçoit les données de S3 et renvoie les résultats à l'utilisateur.

Considérations relatives à S3 Access Grants avec Amazon EMR

Tenez compte des comportements et des limitations suivants lorsque vous utilisez S3 Access Grants avec Amazon EMR.

Prise en charge des fonctionnalités

- S3 Access Grants est pris en charge à partir de la version 6.15.0 d'Amazon EMR.
- Spark est le seul moteur de requêtes pris en charge lorsque vous utilisez S3 Access Grants avec Amazon EMR.
- Delta Lake et Hudi sont les seuls formats de table ouverts pris en charge lorsque vous utilisez S3 Access Grants avec Amazon EMR.
- Les fonctionnalités Amazon EMR suivantes ne sont pas prises en charge pour une utilisation avec S3 Access Grants :
  - Tables Apache Iceberg
  - Authentification native LDAP
  - Authentification native Apache Ranger
  - AWS CLI demandes adressées à Amazon S3 qui utilisent des rôles IAM
  - Accès à S3 via le protocole open source S3A
- L'option `fallbackToIAM` n'est pas prise en charge pour les clusters EMR qui utilisent la propagation d'identité approuvée avec IAM Identity Center.
- S3 Access Grants avec AWS Lake Formation n'est pris en charge qu'avec les clusters Amazon EMR qui s'exécutent sur Amazon EC2.

Considérations comportementales

- L'intégration native d'Apache Ranger à Amazon EMR permet une fonctionnalité semblable à S3 Access Grants dans le cadre du plug-in EMRFS S3 Apache Ranger. Si vous utilisez Apache Ranger pour un contrôle précis des accès (FGAC), nous vous recommandons d'utiliser ce plug-in au lieu de S3 Access Grants.
- Amazon EMR fournit un cache d'informations d'identification dans EMRFS afin de garantir qu'un utilisateur n'a pas besoin de demander à plusieurs reprises les mêmes informations d'identification dans le cadre d'une tâche Spark. Par conséquent, Amazon EMR demande toujours le niveau de privilège par défaut lorsqu'il demande des informations d'identification. Pour plus d'informations, voir la rubrique Demande d'accès aux données S3 du Guide de l'utilisateur Amazon S3.

- Dans le cas où un utilisateur exécute une action non prise en charge par S3 Access Grants, Amazon EMR est configuré pour utiliser le rôle IAM spécifié pour l'exécution de la tâche. Pour plus d'informations, consultez Basculement vers les rôles IAM.

Lancement d'un cluster Amazon EMR avec les octrois d'accès S3

Cette section explique comment lancer un cluster EMR qui s'exécute sur Amazon EC2 et utilise S3 Access Grants pour gérer l'accès aux données dans Amazon S3. Pour savoir comment utiliser S3 Access Grants avec d'autres déploiements Amazon EMR, consultez la documentation suivante :

- Utilisation de S3 Access Grants avec Amazon EMR sur EKS
- Utilisation de S3 Access Grants avec EMR sans serveur

Procédez comme suit pour lancer un cluster EMR qui s'exécute sur Amazon EC2 et utilise les S3 Access Grants pour gérer l'accès aux données dans Amazon S3.

1. Définissez un rôle d'exécution de tâches pour votre cluster EMR. Ajoutez les autorisations IAM `s3:GetDataAccess` et `s3:GetAccessGrantsInstanceForPrefix` requises pour l'exécution des tâches Spark :

```
{
    "Effect": "Allow",
    "Action": [
    "s3:GetDataAccess",
    "s3:GetAccessGrantsInstanceForPrefix"
    ],
    "Resource": [     //LIST ALL INSTANCE ARNS THAT THE ROLE IS ALLOWED TO QUERY
         "arn:aws_partition:s3:Region:account-id1:access-grants/default",
         "arn:aws_partition:s3:Region:account-id2:access-grants/default"
    ]
}
```

> **Note**
>
> Avec Amazon EMR, S3 Access Grants augmente les autorisations définies dans les rôles IAM. Si les rôles IAM que vous spécifiez pour l'exécution des tâches contiennent des autorisations permettant d'accéder directement à S3, les utilisateurs peuvent être

en mesure d'accéder à davantage de données que celles que vous définissez dans S3 Access Grants.

2. Ensuite, utilisez le AWS CLI pour créer un cluster avec Amazon EMR 6.15 ou version ultérieure et la `emrfs-site` classification pour activer les subventions d'accès S3, comme dans l'exemple suivant :

```
aws emr create-cluster
  --release-label emr-6.15.0 \
  --instance-count 3 \
  --instance-type m5.xlarge \
  --configurations '[{"Classification":"emrfs-site",
 "Properties":{"fs.s3.s3AccessGrants.enabled":"true",
 "fs.s3.s3AccessGrants.fallbackToIAM":"false"}}]'
```

Subventions d'accès S3 avec AWS Lake Formation

Si vous utilisez Amazon EMR dans le cadre de l'intégration àAWS Lake Formation, vous pouvez utiliser Amazon S3 Access Grants pour un accès direct ou tabulaire aux données d'Amazon S3.

> **Note**
>
> S3 Access Grants with n' AWS Lake Formation est pris en charge qu'avec les clusters Amazon EMR exécutés sur Amazon EC2.

Accès direct

L'accès direct implique tous les appels pour accéder aux données S3 qui n'appellent pas l'API du service AWS Glue que Lake Formation utilise comme métastore avec Amazon EMR, par exemple, pour appeler : `spark.read`

```
spark.read.csv("s3://...")
```

Lorsque vous utilisez S3 Access Grants AWS Lake Formation sur Amazon EMR, tous les modèles d'accès direct passent par S3 Access Grants pour obtenir des informations d'identification S3 temporaires.

Accès tabulaire

L'accès tabulaire est utilisé lorsque Lake Formation invoque l'API du métastore pour accéder à votre emplacement S3, par exemple, pour interroger les données d'une table :

```
spark.sql("select * from test_tbl")
```

Lorsque vous utilisez S3 Access Grants AWS Lake Formation sur Amazon EMR, tous les modèles d'accès tabulaires passent par Lake Formation.

Basculement vers les rôles IAM

Si un utilisateur tente d'effectuer une action non prise en charge par S3 Access Grants, Amazon EMR utilise par défaut le rôle IAM spécifié pour l'exécution des tâches lorsque le paramètre `fallbackToIAM` est défini sur `true`. Cela permet aux utilisateurs de basculer vers leur rôle d'exécution des tâches pour fournir des informations d'identification pour accéder à S3 dans les scénarios non couverts par S3 Access Grants.

Lorsque le paramètre `fallbackToIAM` est activé, les utilisateurs peuvent accéder aux données autorisées par Access Grants. S'il n'existe pas de jeton S3 Access Grants pour les données concernées, Amazon EMR vérifie l'autorisation nécessaire est accordée son rôle d'exécution des tâches.

> **Note**
>
> Nous vous recommandons de tester vos autorisations d'accès avec le paramètre `fallbackToIAM` activé, même si vous prévoyez de le désactiver pour les charges de travail de production. Pour les tâches Spark, les utilisateurs peuvent accéder à tous les jeux d'autorisations par d'autres moyens à l'aide de leurs informations d'identification IAM. Lorsqu'elles sont activées sur les clusters EMR, les octrois d'accès de S3 permettent aux tâches Spark d'accéder aux emplacements S3. Veillez à protéger ces emplacements S3 contre tout accès en dehors d'EMRFS. Veillez par exemple à protéger les emplacements S3 contre tout accès par des clients S3 utilisés dans les blocs-notes ou par des applications non prises en charge par S3 Access Grants comme Hive ou Presto.

Authentification auprès des nœuds de cluster Amazon EMR

Des clients SSH peuvent utiliser une paire de clés Amazon EC2 pour authentifier des instances de cluster. Avec les versions 5.10.0 et supérieures d'Amazon EMR, vous pouvez également configurer Kerberos pour authentifier les utilisateurs et les connexions SSH auprès du nœud primaire. Et avec les versions 5.12.0 et supérieures d'Amazon EMR, vous pouvez vous authentifier avec LDAP.

Rubriques

- Utilisation d'une paire de clés Amazon EC2 pour les informations d'identification SSH
- Utilisation de Kerberos pour l'authentification avec Amazon EMR
- Utilisation de serveurs Active Directory ou LDAP pour l'authentification avec Amazon EMR

Utilisation d'une paire de clés Amazon EC2 pour les informations d'identification SSH

Les nœuds de cluster Amazon EMR s'exécutent 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 au cluster 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.

- Pour plus d'informations sur la création d'une paire de clés Amazon EC2, consultez les paires de clés Amazon EC2 dans le guide de l'utilisateur Amazon EC2.
- Pour obtenir des instructions sur l'utilisation de PuTTYgen pour créer un fichier .ppk à partir d'un fichier .pem, consultez la section Conversion de votre clé privée à l'aide de PuTTYgen dans le guide de l'utilisateur Amazon EC2.
- Pour plus d'informations sur la définition des autorisations des fichiers .pem et sur la manière de se connecter au nœud principal d'un cluster EMR à l'aide de différentes méthodes, notamment

depuis `ssh` Linux ou macOS, PuTTY depuis Windows ou AWS CLI depuis n'importe quel système d'exploitation compatible, consultez. Connexion au nœud primaire à l'aide de SSH

Utilisation de Kerberos pour l'authentification avec Amazon EMR

Les versions 5.10.0 et supérieures d'Amazon EMR prennent en charge Kerberos. Kerberos est un protocole d'authentification réseau qui utilise la cryptographie à clé secrète pour fournir une authentification forte afin que les mots de passe ou autres informations d'identification ne soient pas envoyés sur le réseau dans un format non chiffré.

Dans Kerberos, les services et utilisateurs qui ont besoin de s'authentifier sont appelés mandataires. Les mandataires existent au sein d'un domaine Kerberos. Dans ce domaine, un serveur Kerberos connu comme le centre de distribution de clés (KDC) fournit aux mandataires les moyens de s'authentifier. Le KDC effectue cette opération en émettant des tickets d'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 informations d'authentification provenant de mandataires d'autres domaines, ce qui est connu sous le nom d'approbation inter-domaines. De plus, un cluster EMR peut utiliser un KDC externe pour authentifier les mandataires.

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 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 n'est pas pris AWS Directory Service for Microsoft Active Directory en charge dans le cadre d'une confiance interdomaines ou en tant que 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

- Applications prises en charge
- Options d'architecture Kerberos
- Configuration de Kerberos sur Amazon EMR
- Utilisation de SSH pour se connecter aux clusters Kerberos
- Didacticiel : configuration d'un KDC dédié au cluster
- Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory

Applications prises en charge

Dans un cluster EMR, les principaux Kerberos sont les services applicatifs 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.

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 les composants pris en charge peuvent varier selon les différentes versions d'Amazon EMR.

L'interface utilisateur Livy est la seule interface utilisateur web hébergée sur le cluster qui est activée pour Kerberos.

- Hadoop MapReduce
- Hbase
- HCatalog
- HDFS

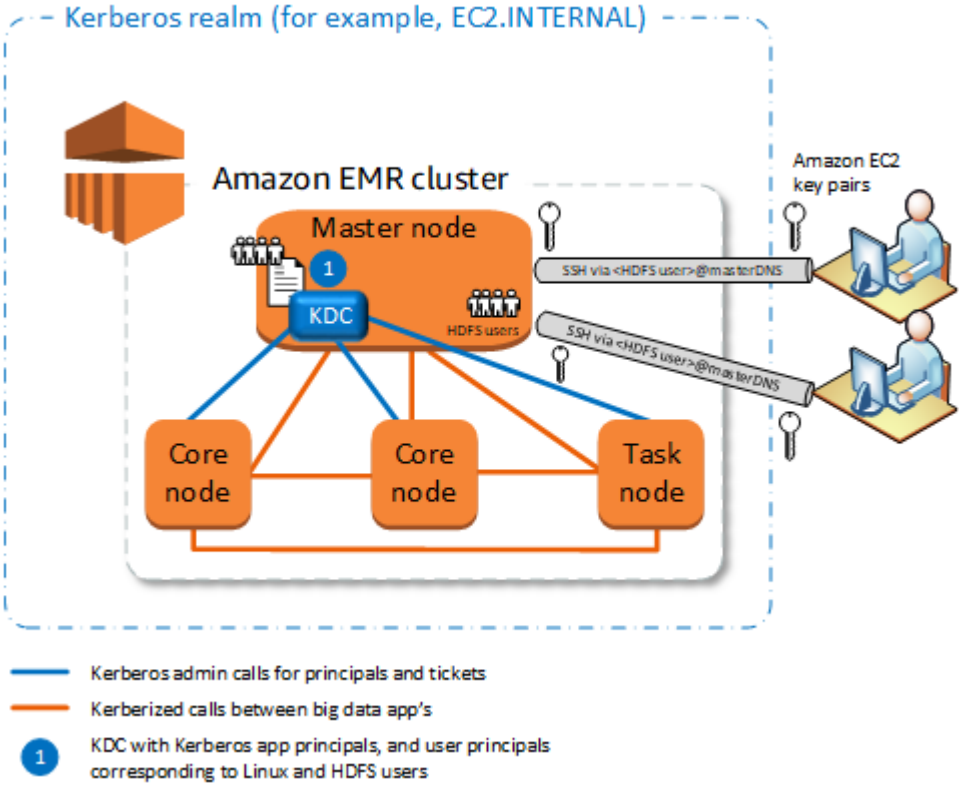
- Hive
  - N'activez pas Hive avec l'authentification LDAP. Cela peut entraîner des problèmes de communication avec Kerberos YARN.
- Hue
  - L'authentification utilisateur Hue n'est pas définie automatiquement et peut être configurée à l'aide de l'API de configuration.
  - Le serveur Hue est activé pour Kerberos. Le serveur frontal Hue (interface utilisateur) n'est pas configuré pour l'authentification. L'authentification LDAP peut être configurée pour l'interface utilisateur Hue.
- Livy
  - L'emprunt d'identité Livy pour les clusters activés pour Kerberos est pris en charge dans les versions 5.22.0 et ultérieures d'Amazon EMR.
- Oozie
- Phoenix
- Presto
  - Presto prend en charge l'authentification Kerberos dans les versions 6.9.0 et supérieures d'Amazon EMR.
  - Pour utiliser l'authentification Kerberos pour Presto, vous devez activer le chiffrement en transit.
- Spark
- Tez
- Trino
  - Trino prend en charge l'authentification Kerberos dans les versions 6.11.0 et ultérieures d'Amazon EMR.
  - Pour utiliser l'authentification Kerberos pour Trino, vous devez activer le chiffrement en transit.
- YARN
- Zeppelin
  - Zeppelin est uniquement configuré pour utiliser Kerberos avec l'interpréteur Spark. Il n'est pas configuré pour les autres interpréteurs.
  - L'emprunt d'identité de l'utilisateur n'est pas pris en charge pour les interpréteurs Zeppelin activés pour Kerberos autres que Spark.
- Zookeeper
  - Le client Zookeeper n'est pas pris en charge.

Options d'architecture Kerberos

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 Configuration de Kerberos sur Amazon EMR.

KDC dédié du cluster (KDC sur le nœud primaire)

Cette configuration est disponible avec les versions 5.10.0 et supérieures d'Amazon EMR.



Avantages

- Amazon EMR possède la propriété totale du KDC.
- Le KDC sur le cluster EMR est indépendant des implémentations KDC centralisées telles que Microsoft Active Directory ou AWS Managed Microsoft AD.

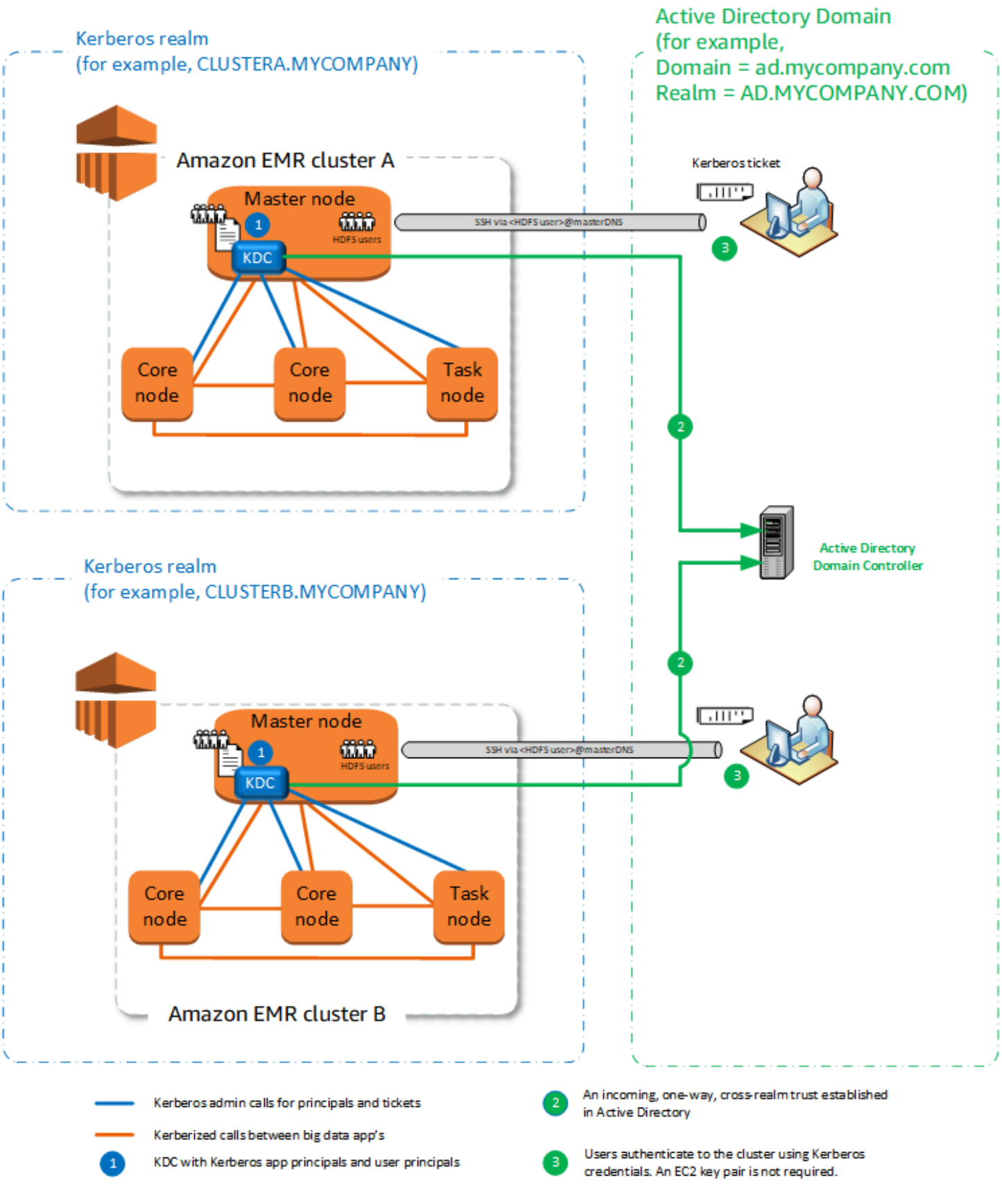
- L'impact de performance est minime, car le KDC gère l'authentification uniquement pour les nœuds locaux au sein du cluster.
- Le cas échéant, d'autres clusters Kerberos peuvent faire référence au KDC comme KDC externe. Pour plus d'informations, consultez KDC externe – Nœud primaire sur un cluster différent.

Considérations et restrictions

- Les clusters activés pour Kerberos ne peuvent pas s'authentifier les uns aux autres. Par conséquent, les applications ne peuvent pas interagir. Si vos applications de cluster ont besoin d'interagir, vous devez établir une approbation inter-domaines entre les clusters ou configurer un cluster comme le KDC externe pour d'autres clusters. Si une approbation inter-domaines est établie, les KDC doivent disposer de différents domaines Kerberos.
- Vous devez créer des utilisateurs Linux sur l'instance EC2 du nœud primaire qui correspondent aux mandataires d'utilisateur KDC, ainsi que les annuaires HDFS pour chaque utilisateur.
- Les mandataires de l'utilisateur doivent utiliser un fichier de clé privée EC2 et les informations d'identification `kinit` pour se connecter au cluster à l'aide de SSH.

Relation d'approbation inter-domaines

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 sur le nœud primaire établit une relation d'approbation avec un autre KDC à l'aide d'un principal inter-domaines qui existe dans les deux KDC. Le nom et le mot de passe du principal 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.


Active Directory Domain
(for example,
Domain = ad.mycompany.com
Realm = AD.MYCOMPANY.COM)
Kerberos realm
(for example, CLUSTERA.MYCOMPANY)
Amazon EMR cluster A
Master node
1
KDC
HDFS users
SSH via <HDFS user>@masterDNS
Kerberos ticket
3
Core node
Core node
Task node
2
Active Directory Domain Controller
Kerberos realm
(for example, CLUSTERB.MYCOMPANY)
Master node
1
KDC
HDFS users
SSH via <HDFS user>@masterDNS
2
3
Core node
Core node
Task node
Amazon EMR cluster B
Kerberos admin calls for principals and tickets
Kerberized calls between big data app's
1
KDC with Kerberos app principals and user principals
2
An incoming, one-way, cross-realm trust established in Active Directory
3
Users authenticate to the cluster using Kerberos credentials. An EC2 key pair is not required.

Avantages

- Le cluster EMR sur lequel le KDC est installé conserve le contrôle total du KDC.
- Avec Active Directory, Amazon EMR crée automatiquement des utilisateurs Linux qui correspondent aux principaux de l'utilisateur provenant du KDC. Vous devez toujours créer des annuaires HDFS pour chaque utilisateur. En outre, les mandataires d'utilisateur dans le domaine Active Directory peuvent accéder aux clusters Kerberos à l'aide des informations d'identification `kinit`, sans le fichier de clé privée EC2. Cela élimine la nécessité de partager le fichier de clé privée entre les utilisateurs de cluster.
- Étant donné que chaque cluster KDC gère l'authentification pour les nœuds du cluster, les effets de la latence du réseau et la surcharge de traitement pour un grand nombre de nœuds dans les clusters est réduit.

Considérations et restrictions

- Si vous établissez une approbation avec un domaine Active Directory, vous devez fournir un nom d'utilisateur et un mot de passe Active Directory avec des autorisations pour joindre des mandataires au domaine lorsque vous créez le cluster.
- Les relations d'approbation inter-domaines ne peuvent pas être établies entre domaines Kerberos portant le même nom.
- Les relations d'approbation inter-domaines doivent être explicitement créées. Par exemple, si les clusters A et B établissent tous deux une relation d'approbation inter-domaines avec un KDC, ils n'ont pas intrinsèquement confiance l'un à l'autre et leurs applications ne peuvent pas s'authentifier pour l'interopérabilité.
- Les KDC doivent être conservés de manière indépendante et coordonnée afin que les informations d'identification des mandataires d'utilisateur correspondent précisément.

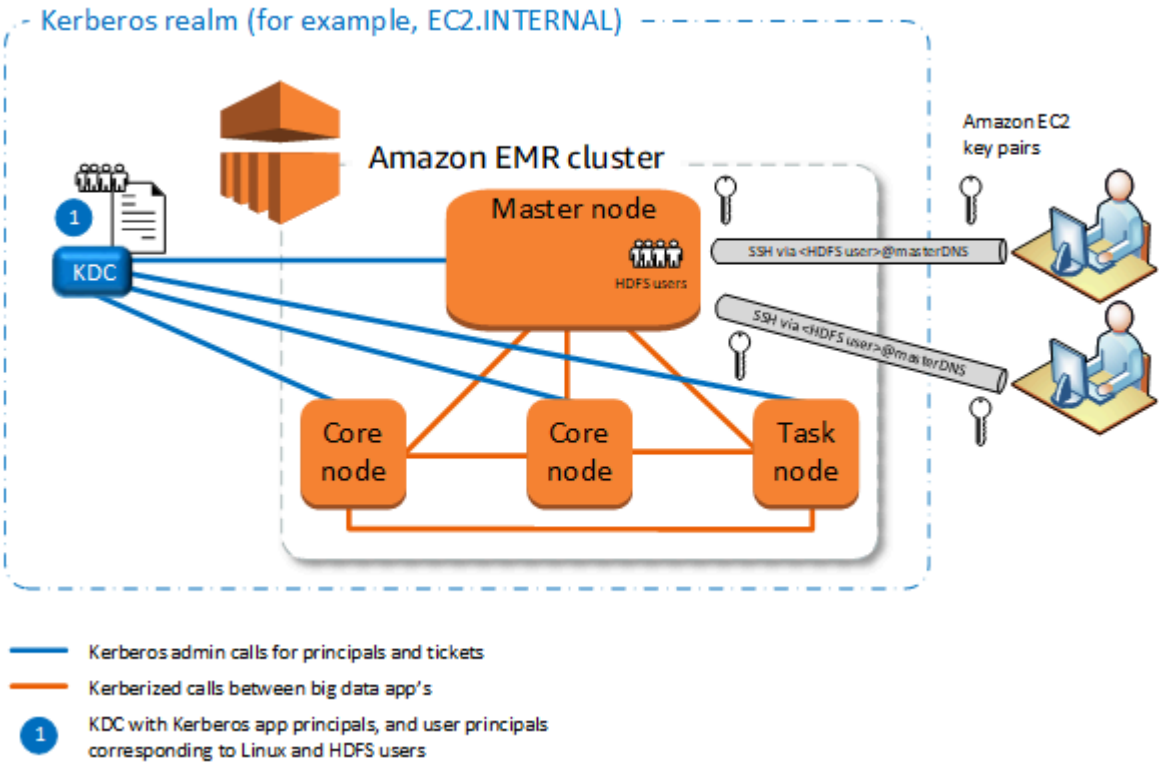
KDC externe

Les configurations avec un KDC externe sont prises en charge avec les versions d'Amazon EMR 5.20.0 et ultérieures.

- KDC externe – MIT KDC
- KDC externe – Nœud primaire sur un cluster différent
- KDC externe – cluster KDC sur un autre cluster avec une relation d'approbation inter-domaines Active Directory

KDC externe – MIT KDC

Cette configuration permet à un ou plusieurs clusters EMR d'utiliser les mandataires définis et maintenus dans un serveur KDC MIT.



Avantages

- La gestion des mandataires est regroupée dans un seul KDC.
- Plusieurs clusters peuvent utiliser le même KDC dans le même domaine Kerberos. Pour plus d'informations, consultez Conditions requises pour l'utilisation de plusieurs clusters avec le même KDC.
- Le nœud primaire sur un cluster activé pour Kerberos ne dispose pas des fardeaux de performances associés à l'entretien du KDC.

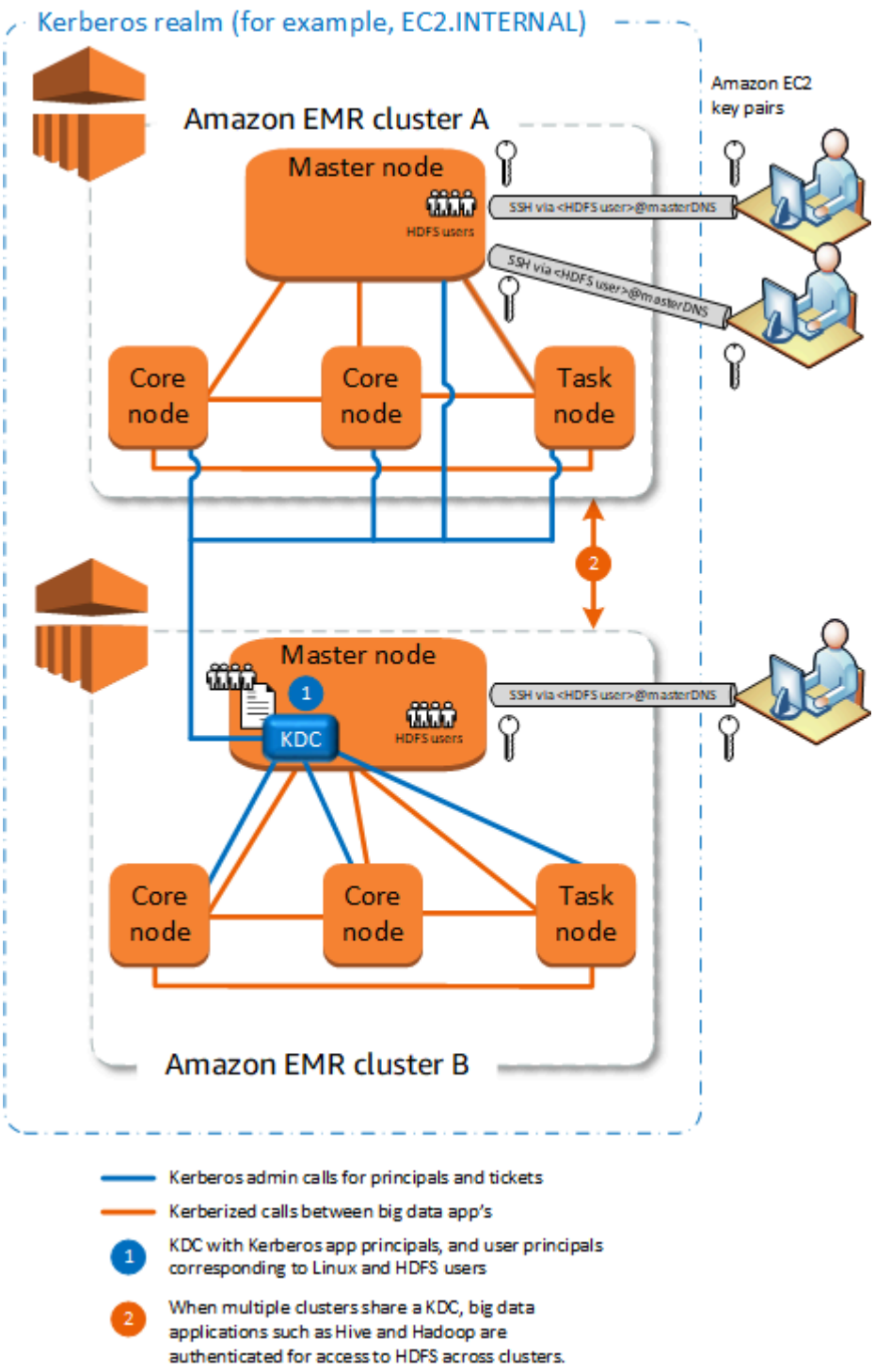
Considérations et restrictions

- Vous devez créer des utilisateurs Linux sur l'instance EC2 du nœud primaire de chaque cluster Kerberos qui correspondent aux mandataires d'utilisateur KDC, ainsi que les annuaires HDFS pour chaque utilisateur.

- Les mandataires de l'utilisateur doivent utiliser un fichier de clé privée EC2 et les informations d'identification `kinit` pour se connecter au cluster Kerberos à l'aide de SSH.
- Chaque nœud activé pour les clusters EMR Kerberos doit avoir un chemin réseau vers le KDC.
- Chaque nœud des clusters activés pour Kerberos passe une charge d'authentification sur le KDC externe et, par conséquent, la configuration du KDC affecte les performances du cluster. Lorsque vous configurez le matériel du serveur KDC, tenez compte du nombre maximal de nœuds Amazon EMR qu'il faut simultanément prendre en charge.
- Les performances du cluster dépendent de la latence de réseau entre les nœuds dans les clusters et le KDC activés pour Kerberos.
- La résolution de problèmes peut être plus difficile en raison d'interdépendances.

KDC externe – Nœud primaire sur un cluster différent

Cette configuration est presque identique à l'implémentation KDC MIT externe ci-dessus, sauf que le KDC est sur le nœud primaire d'un cluster EMR. Pour plus d’informations, consultez KDC dédié du cluster (KDC sur le nœud primaire) et Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.



Avantages

- La gestion des mandataires est regroupée dans un seul KDC.

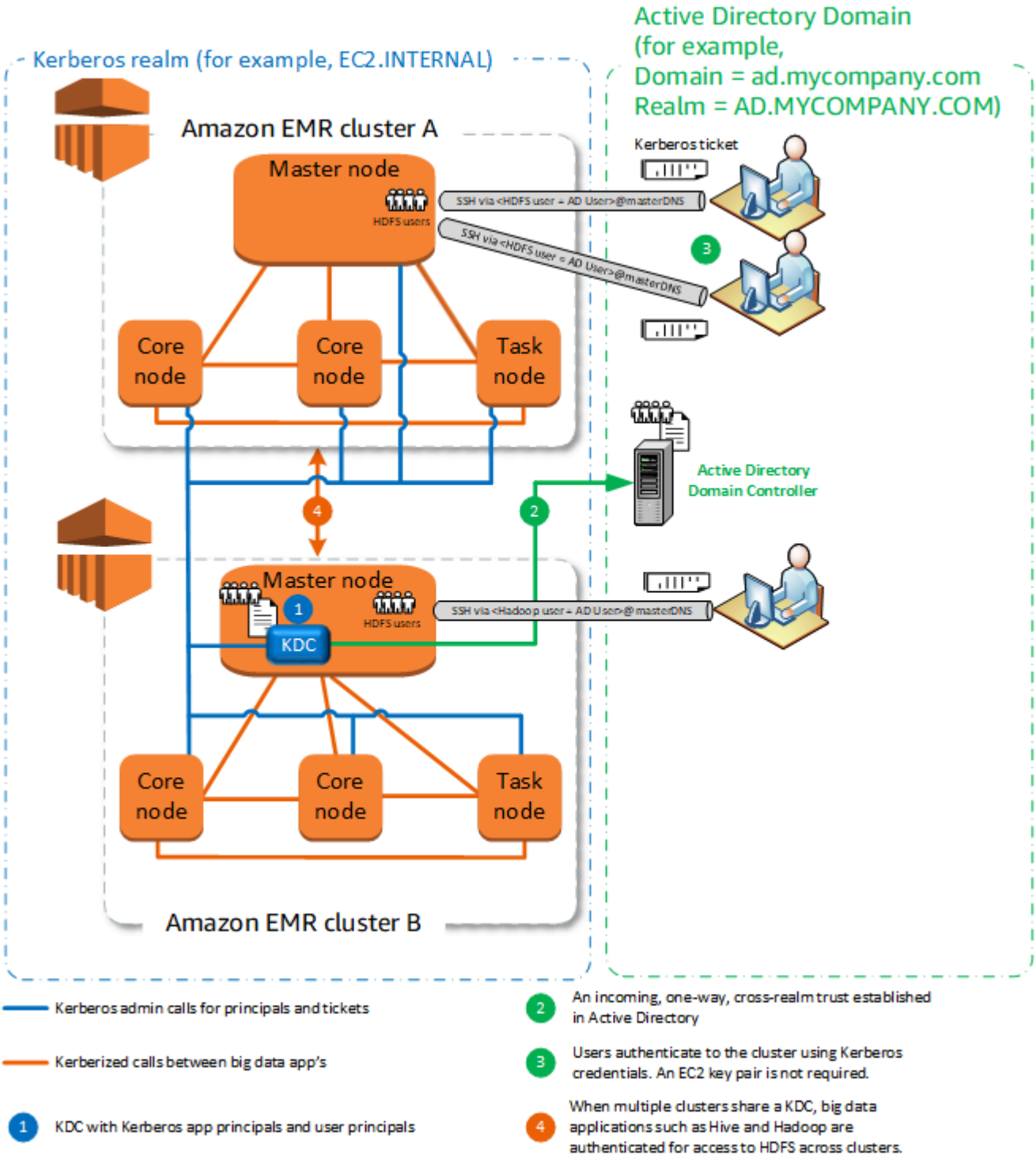
- Plusieurs clusters peuvent utiliser le même KDC dans le même domaine Kerberos. Pour plus d'informations, consultez Conditions requises pour l'utilisation de plusieurs clusters avec le même KDC.

Considérations et restrictions

- Vous devez créer des utilisateurs Linux sur l'instance EC2 du nœud primaire de chaque cluster Kerberos qui correspondent aux mandataires d'utilisateur KDC, ainsi que les annuaires HDFS pour chaque utilisateur.
- Les mandataires de l'utilisateur doivent utiliser un fichier de clé privée EC2 et les informations d'identification `kinit` pour se connecter au cluster Kerberos à l'aide de SSH.
- Chaque nœud dans chaque cluster EMR doit disposer d'un chemin réseau vers le KDC.
- Chaque nœud Amazon EMR des clusters activés pour Kerberos passe une charge d'authentification sur le KDC externe et, par conséquent, la configuration du KDC affecte les performances du cluster. Lorsque vous configurez le matériel du serveur KDC, tenez compte du nombre maximal de nœuds Amazon EMR qu'il faut simultanément prendre en charge.
- Les performances du cluster dépendent de la latence de réseau entre les nœuds dans les clusters et le KDC.
- La résolution de problèmes peut être plus difficile en raison d'interdépendances.

KDC externe – cluster KDC sur un autre cluster avec une relation d'approbation inter-domaines Active Directory

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 Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory. Vous pouvez ensuite lancer d'autres clusters faisant référence au cluster KDC qui a la confiance comme KDC externe. Pour obtenir un exemple, consultez KDC de cluster externe avec approbation inter-domaines Active Directory. 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.



Avantages

- La gestion des mandataires est regroupée dans le domaine Active Directory.

- Amazon EMR se joint au domaine Active Directory, ce qui élimine le besoin de créer des utilisateurs Linux qui correspondent aux utilisateurs Active Directory. Vous devez toujours créer des annuaires HDFS pour chaque utilisateur.
- Plusieurs clusters peuvent utiliser le même KDC dans le même domaine Kerberos. Pour plus d'informations, consultez Conditions requises pour l'utilisation de plusieurs clusters avec le même KDC.
- Les mandataires d'utilisateur dans le domaine Active Directory peuvent accéder aux clusters Kerberos à l'aide des informations d'identification `kinit`, sans le fichier de clé privée EC2. Cela élimine la nécessité de partager le fichier de clé privée entre les utilisateurs de cluster.
- Un seul nœud primaire Amazon EMR a la charge de maintenir le KDC, et seul ce cluster doit être créé avec des informations d'identification Active Directory pour l'approbation inter-domaines entre le KDC et Active Directory.

Considérations et restrictions

- Chaque nœud de chaque cluster EMR doit disposer d'un chemin réseau pour le KDC et le contrôleur de domaine Active Directory.
- Chaque nœud Amazon EMR passe une charge d'authentification sur le KDC externe et, par conséquent, la configuration du KDC affecte les performances du cluster. Lorsque vous configurez le matériel du serveur KDC, tenez compte du nombre maximal de nœuds Amazon EMR qu'il faut simultanément prendre en charge.
- Les performances du cluster dépendent de la latence de réseau entre les nœuds dans les clusters et le serveur KDC.
- La résolution de problèmes peut être plus difficile en raison d'interdépendances.

Conditions requises pour l'utilisation de plusieurs clusters avec le même KDC

Plusieurs clusters peuvent utiliser le même KDC dans le même domaine Kerberos. Toutefois, si les clusters s'exécutent simultanément, ils risquent d'échouer s'ils utilisent des ServicePrincipal noms Kerberos en conflit.

Si vous avez plusieurs clusters simultanés avec le même KDC externe, assurez-vous que les clusters utilisent des domaines Kerberos différents. Si les clusters doivent utiliser le même domaine Kerberos, assurez-vous que les clusters se trouvent dans des sous-réseaux différents et que leurs plages d'adresses CIDR ne se chevauchent pas.

Configuration de Kerberos sur 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.

Étape 1 : Créer une configuration de sécurité avec des propriétés Kerberos

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. Vous pouvez créer une configuration de sécurité à l'aide de la console Amazon EMR, de l'API EMR ou de l' AWS CLI 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. Pour plus d'informations sur les propriétés Kerberos dans une configuration de sécurité, consultez Paramètres Kerberos pour les configurations de sécurité.

Étape 2 : Créer un cluster et spécifier les attributs Kerberos propres au cluster

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'API AWS CLI ou Amazon EMR, assurez-vous de spécifier des options Kerberos compatibles avec la configuration de sécurité spécifiée. Par exemple, si vous spécifiez un mot de passe de principal 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 Paramètres de Kerberos pour les clusters.

Étape 3 : Configurer le nœud primaire de cluster

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 Linux qui correspondent à l'utilisateur Kerberos sont créés. Si les principaux 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 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 Configuration d'un cluster pour les utilisateurs HDFS et les connexions SSH authentifiés par Kerberos.

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 primaire et l'application SSH cliente doit être configurée pour utiliser la GSSAPI. Pour plus d'informations, consultez Configuration d'un cluster pour les utilisateurs HDFS et les connexions SSH authentifiés par Kerberos.

Configuration de sécurité et paramètres de cluster pour Kerberos sur 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.

Paramètres Kerberos pour les configurations de sécurité

Vous pouvez créer une configuration de sécurité qui spécifie les attributs Kerberos à l'aide de la console Amazon EMR, de l'API EMR 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.

Utilisez les références suivantes pour comprendre les paramètres de configuration de sécurité disponibles pour l'architecture Kerberos que vous choisissez. Les paramètres de la console Amazon EMR sont affichés. Pour les options d'interface de ligne de commande correspondantes, consultez Spécification des paramètres Kerberos à l'aide du AWS CLI ou Exemples de configuration.

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 paramètre s Kerberos doivent également être spécifiés sur le cluster, sinon une erreur se produira.
Fournisse ur	KDC dédié du cluster	Spécifie qu'Amazon EMR crée un KDC sur le nœud primaire de tout cluster utilisant cette configuration de sécurité. Vous spécifiez le nom de domaine et le mot de passe administrateur du KDC lorsque vous créez le cluster. Vous pouvez référencer ce KDC à partir d'autres clusters, si nécessaire. 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 du KDC que vous spécifiez pour le KDC dédié au cluster.
	KDC externe	Disponible uniquement avec les versions Amazon EMR 5.20.0 et supérieures. Spécifie que les clusters utilisant cette configuration de sécurité authentifient les principaux Kerberos à l'aide d'un serveur KDC extérieur au cluster. Aucun KDC n'est créé sur le cluster. Lorsque vous créez le cluster, vous spécifiez le nom de domaine et le mot de passe d'administrateur du KDC pour le KDC externe.
Durée de vie du billet		Facultatif. Spécifie la période pendant 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

Paramètre		Description
		cluster via SSH à l'aide d'informations d'identification Kerberos doivent exécuter `kinit` à partir de la ligne de commande du nœud primaire pour renouveler un ticket après son expiration.
Relation d'approbation inter-domaines		Spécifie une confiance inter-domaines entre un KDC dédié au cluster sur des clusters utilisant cette configuration de sécurité et un KDC dans un autre domaine Kerberos. Les principaux (généralement les utilisateurs) d'un autre domaine sont authentifiés auprès des clusters qui utilisent cette configuration. Une configuration supplémentaire dans l'autre domaine Kerberos est requise. Pour plus d'informations, consultez Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.
Propriétés de confiance entre 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, mais en majuscules.
	Domaine	Spécifie le nom de domaine de l'autre domaine de la relation d'approbation.

Paramètre		Description
	Serveur d'administration	Spécifie le FQDN (nom de domaine complet) 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	Spécifie le FQDN (nom de domaine complet) 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		Spécifie que le KDC externe du cluster est utilisé par le cluster.

Paramètre			Description
Propriétés de KDC externe	Serveur d'administration		Spécifie 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		Spécifie 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		Spécifie que l'authentification principale Kerberos est intégrée à un domaine Microsoft Active Directory.
	Propriétés de l'intégration d'Active Directory	Domaine Active Directory	Spécifie le nom de domaine Kerberos du domaine Active Directory. Par convention, les noms de domaine Kerberos sont généralement identiques au nom de domaine, mais en majuscules.
		Domaine Active Directory	Spécifie le nom du domaine Active Directory.

Paramètre			Description
		Serveur Active Directory	Spécifie le nom de domaine complet (FQDN) du contrôleur de domaine Microsoft Active Directory.

Paramètres de Kerberos pour les clusters

Vous pouvez spécifier les paramètres Kerberos lorsque vous créez un cluster à l'aide de la console Amazon EMR, de l'API EMR AWS CLI ou de l'API EMR.

Utilisez les références suivantes pour comprendre les paramètres de configuration de cluster disponibles pour l'architecture Kerberos que vous choisissez. Les paramètres de la console Amazon EMR sont affichés. Pour les options d'interface de ligne de commande correspondantes, consultez Exemples de configuration.

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	Mot de passe utilisé dans le cluster pour `kadmin` ou `kadmin.local` . Ce sont des interfaces de ligne de commande pour le système d'administration Kerberos V5, qui assure la gestion des principaux Kerberos, des stratégies de mot de passe et des fichiers keytab du 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

Paramètre	Description
	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 "Étape 3 : Ajouter des comptes au domaine pour le cluster EMR".
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 "Étape 3 : Ajouter des comptes au domaine pour le cluster EMR".

Exemples de configuration

Les exemples suivants illustrent les configurations de sécurité et les configurations de cluster pour les scénarios courants. AWS CLI les commandes sont affichées par souci de concision.

KDC local

Les commandes suivantes créent un cluster avec un KDC dédié au cluster qui s'exécute sur le nœud primaire. Une configuration supplémentaire du cluster est requise. Pour plus d'informations, consultez Configuration d'un cluster pour les utilisateurs HDFS et les connexions SSH authentifiés par Kerberos.

Créer une configuration de sécurité

```
aws emr create-security-configuration --name LocalKDCSecurityConfig \
```

```
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc",\
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24 }}}}'
```

Créer un cluster

```
aws emr create-cluster --release-label emr-7.1.0 \
--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
```

KDC dédié au cluster avec approbation inter-domaines Active Directory

Les commandes suivantes créent un cluster avec un KDC dédié au cluster qui s'exécute sur le nœud primaire 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 Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.

Créer une configuration de sécurité

```
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"}}}}}'
```

Créer un cluster

```
aws emr create-cluster --release-label emr-7.1.0 \
--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
```

KDC externe sur un cluster différent

Les commandes suivantes créent un cluster qui fait référence à un KDC dédié au cluster sur le nœud primaire d'un cluster différent pour authentifier les mandataires. Une configuration supplémentaire du cluster est requise. Pour plus d'informations, consultez Configuration d'un cluster pour les utilisateurs HDFS et les connexions SSH authentifiés par Kerberos.

Créer une configuration de sécurité

```
aws emr create-security-configuration --name ExtKDCOnDifferentCluster \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSOfKDCMaster:749", \
"KdcServer": "MasterDNSOfKDCMaster:88"}}}}'
```

Créer un cluster

```
aws emr create-cluster --release-label emr-7.1.0 \
--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
```

KDC de cluster externe avec approbation inter-domaines Active Directory

Les commandes suivantes créent un cluster sans KDC. Le cluster fait référence à un KDC dédié au cluster s'exécutant sur le nœud primaire 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 primaire avec le KDC. Pour plus d'informations, consultez Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory.

Créer une configuration de sécurité

```
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", \
"AdServer":"ad.domain.com"}}}}}'
```

Créer un cluster

```
aws emr create-cluster --release-label emr-7.1.0 \
--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
```

Configuration d'un cluster pour les utilisateurs HDFS et les connexions SSH authentifiés par Kerberos

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 :

- Un compte Linux correspondant au principal Kerberos dans le KDC doit exister sur le cluster. Amazon EMR le fait automatiquement dans les architectures qui s'intègrent à Active Directory.
- Vous devez créer un répertoire d'utilisateurs HDFS sur le nœud primaire pour chaque utilisateur et donner à l'utilisateur les autorisations pour le répertoire.
- Vous devez configurer le service SSH afin que la GSSAPI soit activée sur le nœud primaire. De plus, les utilisateurs doivent disposer d'un client SSH avec la GSSAPI activée.

Ajout d'utilisateurs Linux et de mandataires Kerberos au nœud primaire

Si vous n'utilisez pas Active Directory, vous devez créer des comptes Linux sur le nœud primaire du cluster et ajouter des mandataires au KDC pour ces utilisateurs Linux. Cela comprend un principal dans le KDC pour le nœud primaire. En plus des principaux d'utilisateurs, le KDC qui s'exécute sur le nœud primaire 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

plus d'informations, consultez Relation d'approbation inter-domaines et KDC externe – cluster KDC sur un autre cluster avec une relation d'approbation inter-domaines Active Directory.

> **Important**
>
> Le KDC, ainsi que la base de données des principaux, sont perdus lorsque le nœud primaire est résilié, car ce dernier utilise un stockage éphémère. Si vous créez des utilisateurs pour les connexions SSH, nous vous recommandons d'établir une confiance interdomaines avec un KDC externe configuré pour la haute disponibilité. Par ailleurs, si vous créez des utilisateurs pour les connexions SSH à l'aide de comptes Linux, automatisez le processus de création de compte à l'aide d'actions et de scripts d'amorçage afin de pouvoir le répéter lors de la création d'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 primaire 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 primaire à l'aide de SSH.

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-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,\
Args=["s3://DOC-EXAMPLE-BUCKET/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 primary node, using the primary 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 primary 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 systemctl restart sshd
```

Ajout de répertoires HDFC d'utilisateurs

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 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 primaire 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 primaire à l'aide de SSH.

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 <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-
EXAMPLE-BUCKET/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
```

Activation de GSSAPI pour SSH

Afin de permettre aux utilisateurs authentifiés par Kerberos de se connecter au nœud primaire 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 primaire 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 systemctl restart sshd
```

Utilisation de SSH pour se connecter aux clusters Kerberos

Cette section illustre les étapes pour qu'un utilisateur authentifié par Kerberos se connecte au nœud primaire 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

primaire. 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 à un cluster.

SSH utilise GSSAPI pour authentifier les clients Kerberos et vous devez activer l'authentification GSSAPI pour le service SSH sur le nœud primaire du cluster. Pour plus d'informations, consultez Activation de GSSAPI pour SSH. Les clients SSH doivent également utiliser GSSAPI.

Dans les exemples suivants, pour le MasterPublicDNS, utilisez la valeur qui apparaît pour le ***DNS public principal*** *dans l'onglet* ***Résumé*** *du volet des détails du cluster, par exemple, ec2-11-222-33-44.compute-1.amazonaws.com.*

Conditions préalables pour 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 primaire du cluster.

Pour copier le fichier krb5.conf

1. Utilisez SSH pour vous connecter au nœud primaire à l'aide d'une paire de clés EC2 et de l'utilisateur `hadoop` par défaut. Par exemple, `hadoop@`*`MasterPublicDNS`*. Pour obtenir des instructions complètes, veuillez consulter Connexion à un cluster.
2. Dans le nœud primaire, copiez le contenu du fichier `/etc/krb5.conf`. Pour plus d'informations, consultez Connexion à un cluster.
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.

Utilisation de Kinit et 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 principal 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.

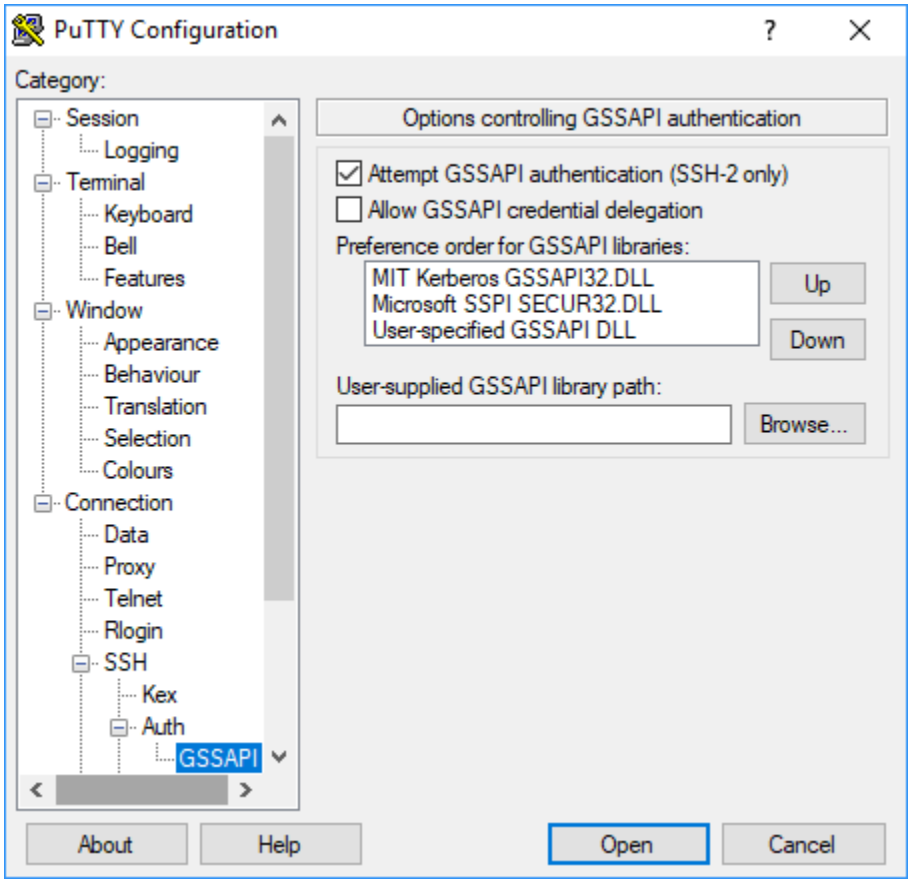
   Exemple : utilisateurs Linux

   L'option `-K` spécifie l'authentification GSSAPI.

   ```
   ssh -K user1@MasterPublicDNS
   ```

   Exemple : utilisateurs Windows (PuTTY)

   Assurez-vous que l'authentification GSSAPI est activée pour la session, telle qu'illustrée :



Didacticiel : configuration d'un KDC dédié au cluster

Cette rubrique vous guide dans la création d'un cluster avec un centre de diffusion d'une clé (KDC) dédié au cluster, dans l'ajout manuel de comptes Linux à tous les nœuds du cluster, dans l'ajout de principaux Kerberos au KDC sur le nœud primaire et dans la vérification de l'installation d'un client Kerberos sur les ordinateurs clients.

Pour plus d'informations sur la prise en charge de Kerberos et de KDC par Amazon EMR, ainsi que des liens vers la documentation MIT Kerberos, consultez Utilisation de Kerberos pour l'authentification avec Amazon EMR.

Étape 1 : Créer le cluster activé pour Kerberos

1. Créez une configuration de sécurité qui active Kerberos. L'exemple suivant illustre une `create-security-configuration` commande utilisant le AWS CLI qui spécifie la configuration de sécurité sous la forme d'une structure JSON intégrée. 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 fait référence à la configuration de sécurité que vous avez créée ci-dessus, `MyKerberosConfig`. Elle fait également référence à un script simple, `createlinuxusers.sh`, sous forme d'action d'amorçage, que vous créez et chargez dans Amazon S3 avant la création du cluster.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-7.1.0 \
--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://DOC-EXAMPLE-BUCKET/createlinuxusers.sh
```

Le code 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
```

Étape 2 : Ajouter des principaux au KDC, créer des répertoires d'utilisateurs HDFS et configurer SSH

Le KDC qui s'exécute sur le nœud primaire 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 primaire 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-2AL4XXXXXX5T9> --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myregion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-
BUCKET/configurekdc.sh"]
```

Le code suivant illustre le contenu du script `configurekdc.sh`.

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary 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 primary 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 systemctl restart sshd
```

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 Utilisation de SSH pour se connecter aux clusters Kerberos.

Didacticiel : configuration d'une approbation inter-domaines avec un domaine Active Directory

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 centre de diffusion d'une clé (KDC) dédié au cluster établit une relation d'approbation avec un autre KDC à l'aide d'un principal interdomaines qui existe dans les deux KDC. 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 KDC de cluster externe avec approbation inter-domaines Active Directory.

Pour plus d'informations sur la prise en charge de Kerberos et de KDC par Amazon EMR, ainsi que des liens vers la documentation MIT Kerberos, consultez Utilisation de Kerberos pour l'authentification avec Amazon EMR.

> **Important**
>
> Amazon EMR ne prend pas en charge les approbations entre domaines avec. AWS Directory Service for Microsoft Active Directory

Étape 1 : Configuration du VPC et du sous-réseau

Étape 2 : Lancer et installer le contrôleur de domaine Active Directory

Étape 3 : Ajouter des comptes au domaine pour le cluster EMR

Étape 4 : Configurer une approbation entrante sur le contrôleur de domaine Active Directory

Étape 5 : Utiliser un groupe d'options DHCP pour spécifier le contrôleur de domaine Active Directory en tant que serveur DNS de VPC

Étape 6 : Lancer un cluster EMR activé pour Kerberos

Étape 7 : Créer des utilisateurs HDFS et définir des autorisations sur le cluster pour les comptes Active Directory

Étape 1 : Configuration du VPC et du sous-réseau

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 Étape 5 : Utiliser un groupe d'options DHCP pour spécifier le contrôleur de domaine Active Directory en tant que serveur DNS de VPC.

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 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 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, consultez Étape 1 : Créer le VPC dans le Guide de démarrage Amazon VPC.

   > **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 est dû au fait que les noms DNS des ordinateurs du cluster sont utilisés lorsque les ordinateurs rejoignent le répertoire Active Directory. AWS attribue des noms d'hôte DNS en fonction de l'adresse IPv4 de telle sorte que des 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 de plus amples informations, veuillez consulter 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.

Étape 2 : Lancer et installer le contrôleur de domaine Active Directory

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 la section Lancement d'une AWS Marketplace instance dans le guide de l'utilisateur Amazon EC2.
2. Notez l'ID de groupe du groupe de sécurité associé à l'instance EC2. Vous en avez besoin pour Étape 6 : Lancer un cluster EMR activé pour Kerberos. 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, veuillez consulter la section Amazon EC2 security groups for Linux instances (français non garanti) dans le Guide de l'utilisateur Amazon EC2.

3. Connectez-vous à l'instance EC2 à l'aide de RDP. Pour plus d'informations, consultez la section Connexion à votre instance Windows dans le guide de l'utilisateur Amazon EC2.
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.

Étape 3 : Ajouter des comptes au domaine pour le cluster EMR

RDP sur le contrôleur de domaine Active Directory pour créer des comptes dans les utilisateurs et ordinateurs Active Directory pour chaque utilisateur de cluster. Pour plus d'informations, consultez la section Création d'un compte d'utilisateur dans Utilisateurs et ordinateurs Active Directory sur le site Microsoft Learn. Notez le nom de connexion de chaque utilisateur. Vous en aurez besoin ultérieurement lorsque vous configurez le cluster.

En outre, créez un compte 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 Étape 6 : Lancer un cluster EMR activé pour Kerberos. Pour déléguer des privilèges de jointure d'ordinateur au compte, 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 d'annuaire dans le Guide d'administration AWS Directory Service .

Étape 4 : Configurer une approbation entrante sur le contrôleur de domaine Active Directory

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*. Remplacez le *KDC-FQDN* par le nom DNS public répertorié pour le nœud primaire Amazon EMR hébergeant le KDC. Le paramètre `passwordt` spécifie le mot de passe du principal 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
```

Étape 5 : Utiliser un groupe d'options DHCP pour spécifier le contrôleur de domaine Active Directory en tant que serveur DNS de VPC

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 les serveurs de noms de domaine, vous devez spécifier l'adresse IP du contrôleur de domaine Active Directory (qui doit être accessible depuis le cluster) comme première entrée, suivie du AmazonProvidedDNS (par exemple, ***`xx.xx.xx.xx`***, DNS). AmazonProvided Pour plus d'informations, consultez Modification des jeux d'options DHCP.

Étape 6 : Lancer un cluster EMR activé pour Kerberos

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 Étape 2 : Lancer et installer le contrôleur de domaine Active Directory.

   ```
   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 :

   - Utilisez l'option `--security-configuration` pour spécifier la configuration de sécurité que vous avez créée. Nous utilisons *`MyKerberosConfig`* dans l'exemple.
   - Utilisez la propriété `SubnetId` de l'`--ec2-attributes option` pour spécifier le sous-réseau que vous avez créé dans Étape 1 : Configuration du VPC et du sous-réseau. Dans l'exemple, nous utilisons *`step1-subnet`*.
   - Utilisez `AdditionalMasterSecurityGroups` et `AdditionalSlaveSecurityGroups` de l'option `--ec2-attributes` pour spécifier que le groupe de sécurité associé au contrôleur de domaine AD de Étape 2 : Lancer et installer le contrôleur de domaine Active Directory est associé au nœud primaire du cluster, ainsi qu'aux nœuds principaux et de tâches. Dans l'exemple, nous utilisons *`sg-012xrlmdomain345`*.

   Utilisez `--kerberos-attributes` pour spécifier les attributs Kerberos suivants spécifiques au cluster :

   - Le domaine du cluster que vous avez spécifié lors de la configuration du contrôleur de domaine Active Directory.
   - Le mot de passe du principal d'approbation inter-domaines que vous avez spécifié sous la forme `passwordt` dans Étape 4 : Configurer une approbation entrante sur le contrôleur de domaine Active Directory.
   - Un `KdcAdminPassword`, que vous pouvez utiliser pour gérer le KDC dédié au cluster.
   - Le nom de connexion et le mot de passe utilisateur du compte Active Directory avec des privilèges de jointure d'ordinateur que vous avez créé dans Étape 3 : Ajouter des comptes au domaine pour le cluster EMR.

   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
```

Étape 7 : Créer des utilisateurs HDFS et définir des autorisations sur le cluster pour les comptes Active Directory

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 Active Directory. Par exemple, le nom de connexion utilisateur `LiJuan` dans Active Directory dispose d'un compte Linux `lijuan`. Les noms d'utilisateurs Active Directory peuvent contenir des lettres majuscules, mais Linux ne prend pas en charge la casse 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 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 primaire. Utilisez la paire de clés EC2 que vous avez spécifiée lors de la création du cluster pour vous connecter au nœud primaire via SSH en tant qu'utilisateur Hadoop. Pour plus d'informations, consultez Utilisation d'une paire de clés Amazon EC2 pour les informations d'identification SSH.

Exécutez la commande suivante pour ajouter une étape au cluster qui exécute un script, *AddHDFSUsers.sh*.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-BUCKET/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
```

Groupes Active Directory mappés aux groupes Hadoop

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 primaire telle que décrite dans Utilisation de SSH pour se connecter aux clusters Kerberos, 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 consulter d'autres mappages de groupe d'utilisateurs en spécifiant un ou plusieurs noms d'utilisateur avec la commande, par exemple `hdfs groups` *`lijuan`*. Pour de plus amples informations, veuillez consulter groupes dans le Guide de commandes HDFS d'Apache.

Utilisation de serveurs Active Directory ou LDAP pour l'authentification avec Amazon EMR

Avec les versions 6.12.0 et supérieures d'Amazon EMR, vous pouvez utiliser le protocole LDAP sur SSL (LDAPS) pour lancer un cluster qui s'intègre nativement au serveur d'identité de votre entreprise. LDAP (Lightweight Directory Access Protocol) est un protocole d'application ouvert et neutre qui permet d'accéder à des données et de les conserver. LDAP est couramment utilisé pour l'authentification des utilisateurs par rapport aux serveurs d'identité d'entreprise hébergés sur des applications telles que Active Directory (AD) et OpenLDAP. Grâce à cette intégration native, vous pouvez utiliser votre serveur LDAP pour authentifier les utilisateurs sur Amazon EMR.

Les éléments principaux de l'intégration LDAP d'Amazon EMR sont les suivants :

- Amazon EMR configure les applications prises en charge pour qu'elles s'authentifient en votre nom à l'aide de l'authentification LDAP.

- Amazon EMR configure et maintient la sécurité des applications prises en charge avec le protocole Kerberos. Vous n'avez pas besoin de saisir de commandes ou de scripts.
- Vous bénéficiez d'un contrôle précis des accès (FGAC) via l'autorisation Apache Ranger pour la base de données et les tables Hive Metastore. Pour plus d'informations, consultez Intégration d'Amazon EMR avec Apache Ranger.
- Lorsque vous avez besoin d'informations d'identification LDAP pour accéder à un cluster, vous bénéficiez d'un contrôle précis des accès (FGAC) sur les personnes qui peuvent accéder à vos clusters EMR via SSH.

Les pages suivantes présentent une vue d'ensemble conceptuelle, les conditions préalables et les étapes à suivre pour lancer un cluster EMR avec l'intégration LDAP d'Amazon EMR.

Rubriques

- Présentation de LDAP avec Amazon EMR
- Composants LDAP pour Amazon EMR
- Prise en charge des applications et considérations relatives à LDAP pour Amazon EMR
- Configuration et lancement d'un cluster EMR avec LDAP
- Exemples d'utilisation de LDAP avec Amazon EMR

Présentation de LDAP avec Amazon EMR

LDAP (Lightweight Directory Access Protocol) est un protocole logiciel que les administrateurs réseau utilisent pour gérer et contrôler l'accès aux données en authentifiant les utilisateurs au sein du réseau d'une entreprise. Le protocole LDAP stocke les informations dans une structure d'annuaire hiérarchique et arborescente. Pour plus d'informations, consultez la section Concepts de base du protocole LDAP sur LDAP.com.

Au sein du réseau d'une entreprise, de nombreuses applications peuvent utiliser le protocole LDAP pour authentifier les utilisateurs. Avec l'intégration LDAP d'Amazon EMR, les clusters EMR peuvent utiliser nativement le même protocole LDAP avec une configuration de sécurité supplémentaire.

Amazon EMR prend en charge deux implémentations majeures du protocole LDAP : Active Directory et OpenLDAP. Bien que d'autres implémentations soient possibles, la plupart d'entre elles s'adaptent aux mêmes protocoles d'authentification qu'Active Directory ou OpenLDAP.

Active Directory (AD)

Active Directory (AD) est un service d'annuaire de Microsoft pour les réseaux de domaines Windows. AD est inclus dans la plupart des systèmes d'exploitation Windows Server et peut communiquer avec les clients via les protocoles LDAP et LDAPS. Pour l'authentification, Amazon EMR tente d'établir une liaison utilisateur avec votre instance AD en utilisant le nom d'utilisateur principal (UPN) comme nom distinctif et mot de passe. L'UPN utilise le format standard `username@domain_name`.

OpenLDAP

OpenLDAP est une implémentation gratuite et open source du protocole LDAP. Pour l'authentification, Amazon EMR tente une liaison utilisateur avec votre instance OpenLDAP avec le nom de domaine entièrement qualifié (FQDN) comme nom distinctif et mot de passe. Le FQDN utilise le format standard `username_attribute=username,LDAP_user_search_base`. En général, la valeur `username_attribute` est `uid`, et la valeur `LDAP_user_search_base` contient les attributs de l'arbre qui mène à l'utilisateur. Par exemple, `ou=People,dc=example,dc=com`.

D'autres implémentations libres et open source du protocole LDAP suivent généralement un FQDN similaire à celui d'OpenLDAP pour les noms distinctifs de leurs utilisateurs.

Composants LDAP pour Amazon EMR

Vous pouvez utiliser votre serveur LDAP pour vous authentifier auprès d'Amazon EMR et de toutes les applications que l'utilisateur utilise directement sur le cluster EMR grâce aux composants suivants.

Agent secret

L'agent secret est un processus intégré au cluster qui authentifie toutes les demandes des utilisateurs. L'agent secret crée le lien utilisateur vers votre serveur LDAP pour le compte des applications prises en charge sur le cluster EMR. L'agent secret s'exécute en tant qu'utilisateur `emrsecretagent` et écrit des journaux dans le répertoire `/emr/secretagent/log`. Ces journaux fournissent des détails sur l'état de la demande d'authentification de chaque utilisateur et sur les erreurs susceptibles de survenir lors de l'authentification de l'utilisateur.

Démon des services de sécurité du système (SSSD)

SSSD est un démon qui s'exécute sur chaque nœud d'un cluster EMR compatible LDAP. SSSD crée et gère un utilisateur UNIX pour synchroniser votre identité d'entreprise distante avec chaque nœud. Les applications basées sur Yarn telles que Hive et Spark nécessitent qu'un utilisateur UNIX local existe sur chaque nœud qui exécute une requête pour un utilisateur.

Prise en charge des applications et considérations relatives à LDAP pour Amazon EMR

Applications prises en charge avec LDAP pour Amazon EMR

> **Important**
>
> Les applications répertoriées sur cette page sont les seules applications prises en charge par Amazon EMR pour LDAP. Pour garantir la sécurité du cluster, vous ne pouvez inclure des applications compatibles LDAP que lorsque vous créez un cluster EMR avec LDAP activé. Si vous tentez d'installer d'autres applications non prises en charge, Amazon EMR rejettera votre demande de nouveau cluster.

Les versions 6.12 et supérieures d'Amazon EMR prennent en charge l'intégration LDAP avec les applications suivantes :

- Apache Livy
- Apache Hive jusqu'à HiveServer 2 (HS2)
- Trino
- Presto
- Hue

Vous pouvez également installer les applications suivantes sur un cluster EMR et les configurer pour répondre à vos besoins en matière de sécurité :

- Apache Spark
- Apache Hadoop

Fonctionnalités prises en charge avec LDAP pour Amazon EMR

Vous pouvez utiliser les fonctionnalités Amazon EMR avec l'intégration LDAP :

> **Note**
>
> Pour garantir la sécurité des informations d'identification LDAP, vous devez utiliser le chiffrement en transit pour sécuriser le flux de données à destination et en provenance du

cluster. Pour plus d'informations sur le chiffrement en transit, consultez Chiffrer les données au repos et en transit.

- Chiffrement en transit (obligatoire) et au repos
- Groupes d'instances, parcs d'instances et instances Spot
- Reconfiguration des applications sur un cluster en cours d'exécution
- chiffrement côté serveur (SSE) EMRFS

Fonctions non prises en charge

Tenez compte des limites suivantes lorsque vous utilisez l'intégration Amazon EMR LDAP :

- Amazon EMR désactive les étapes pour les clusters sur lesquels le protocole LDAP est activé.
- Amazon EMR ne prend pas en charge les rôles d'exécution ni les AWS Lake Formation intégrations pour les clusters sur lesquels LDAP est activé.
- Amazon EMR ne prend pas en charge le protocole LDAP avec StartTLS.
- Amazon EMR ne prend pas en charge le mode haute disponibilité (clusters avec plusieurs nœuds primaires) pour les clusters sur lesquels LDAP est activé.
- Vous ne pouvez pas faire pivoter les informations d'identification ou les certificats de liaison pour les clusters sur lesquels LDAP est activé. Si l'un de ces champs a fait l'objet d'une rotation, nous vous recommandons de démarrer un nouveau cluster avec les informations d'identification ou les certificats de liaison mis à jour.
- Vous devez utiliser des bases de recherche exactes avec LDAP. La base de recherche d'utilisateurs et de groupes LDAP ne prend pas en charge les filtres de recherche LDAP.

Configuration et lancement d'un cluster EMR avec LDAP

Cette section explique comment configurer Amazon EMR pour une utilisation avec l'authentification LDAP.

Rubriques

- Ajouter AWS Secrets Manager des autorisations au rôle d'instance Amazon EMR
- Création de la configuration de sécurité Amazon EMR pour l'intégration LDAP
- Lancement d'un cluster EMR qui s'authentifie auprès de LDAP

Ajouter AWS Secrets Manager des autorisations au rôle d'instance Amazon EMR

Amazon EMR utilise un rôle de service IAM pour effectuer des actions en votre nom afin d'allouer et de gérer les clusters. 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 spécial qu'Amazon EMR attribue à chaque instance EC2 d'un cluster au moment du lancement.

Pour définir les autorisations permettant à un cluster EMR d'interagir avec les données Amazon S3 et d'autres services AWS , définissez un profil d'instance Amazon EC2 personnalisé au lieu de `EMR_EC2_DefaultRole` lorsque vous lancez votre cluster. Pour plus d'informations, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2) et Personnaliser les rôles IAM.

Ajoutez les instructions suivantes au profil d'instance EC2 par défaut pour permettre à Amazon EMR de baliser les sessions et d'accéder à celles qui stockent AWS Secrets Manager les certificats LDAP.

```
    {
      "Sid": "AllowAssumeOfRolesAndTagging",
      "Effect": "Allow",
      "Action": ["sts:TagSession", "sts:AssumeRole"],
      "Resource": [
        "arn:aws:iam::111122223333:role/LDAP_DATA_ACCESS_ROLE_NAME",
        "arn:aws:iam::111122223333:role/LDAP_USER_ACCESS_ROLE_NAME"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [
            "arn:aws:secretsmanager:us-east-1:111122223333:secret:LDAP_SECRET_NAME*",
            "arn:aws:secretsmanager:us-
east-1:111122223333:secret:ADMIN_LDAP_SECRET_NAME*"
        ]
    }
```

> **Note**
>
> Vos demandes de cluster échoueront si vous oubliez le caractère générique * à la fin du nom du secret lorsque vous définissez les autorisations de Secrets Manager. Le caractère générique représente les versions secrètes.

Vous devez également limiter le champ d'application de la AWS Secrets Manager politique aux seuls certificats dont votre cluster a besoin pour provisionner des instances.

Création de la configuration de sécurité Amazon EMR pour l'intégration LDAP

Avant de lancer un cluster EMR avec intégration LDAP, suivez les étapes dans Création d'une configuration de sécurité pour créer une configuration de sécurité Amazon EMR pour le cluster. Complétez les configurations suivantes dans le bloc `LDAPConfiguration` sous `AuthenticationConfiguration` ou dans les champs correspondants de la section Configurations de sécurité de la console Amazon EMR :

`EnableLDAPAuthentication`

Option de console : Protocole d'authentification : LDAP

Pour utiliser l'intégration LDAP, définissez cette option sur `true` ou sélectionnez-la comme protocole d'authentification lorsque vous créez un cluster dans la console. Par défaut, `EnableLDAPAuthentication` est `true` lorsque vous créez une configuration de sécurité dans la console Amazon EMR.

`LDAPServerURL`

Option de console : Emplacement du serveur LDAP

L'emplacement du serveur LDAP, y compris le préfixe : `ldaps://`*`location_of_server`*.

`BindCertificateARN`

Option de console : Certificat SSL LDAP

L' AWS Secrets Manager ARN qui contient le certificat pour signer le certificat SSL utilisé par le serveur LDAP. Si votre serveur LDAP est signé par une autorité de certification (CA) publique, vous pouvez fournir un AWS Secrets Manager ARN avec un fichier vide. Pour plus d'informations sur le stockage de votre certificat dans Secrets Manager, consultez Stockez les certificats TLS dans AWS Secrets Manager..

`BindCredentialsARN`

Option de console : Informations d'identification de liaison du serveur LDAP

Un AWS Secrets Manager ARN qui contient les informations d'identification de liaison utilisateur de l'administrateur LDAP. Les informations d'identification sont stockées

sous forme d'objet JSON. Il n'y a qu'une seule paire clé-valeur dans ce secret. La clé de la paire est le nom d'utilisateur et la valeur est le mot de passe. Par exemple, `{"uid=admin,cn=People,dc=example,dc=com": "AdminPassword1"}`. Ce champ est facultatif, sauf si vous activez la connexion SSH pour votre cluster EMR. Dans de nombreuses configurations, les instances Active Directory nécessitent des informations d'identification de liaison pour permettre à SSSD de synchroniser les utilisateurs.

`LDAPAccessFilter`

Option de console : Filtre d'accès LDAP

Spécifie le sous-ensemble d'objets de votre serveur LDAP qui peuvent s'authentifier. Par exemple, si vous souhaitez uniquement accorder l'accès à tous les utilisateurs de la classe d'objet `posixAccount` de votre serveur LDAP, définissez le filtre d'accès comme `(objectClass=posixAccount)`.

`LDAPUserSearchBase`

Option de console : Base de recherche d'utilisateurs LDAP

La base de recherche à laquelle appartiennent vos utilisateurs au sein de votre serveur LDAP. Par exemple, `cn=People,dc=example,dc=com`.

`LDAPGroupSearchBase`

Option de console : base de recherche de groupes LDAP

La base de recherche à laquelle appartiennent vos groupes au sein de votre serveur LDAP. Par exemple, `cn=Groups,dc=example,dc=com`.

`EnableSSHLogin`

Option de console : Connexion SSH

Spécifie s'il faut autoriser ou non l'authentification par mot de passe avec les informations d'identification LDAP. Nous vous déconseillons d'activer cette option. Les paires de clés constituent une voie plus sécurisée pour autoriser l'accès aux clusters EMR. Ce champ est facultatif et contient `false` par défaut.

`LDAPServerType`

Option de console : Type de serveur LDAP

Spécifie le type de serveur LDAP auquel Amazon EMR se connecte. Les options prises en charge sont Active Directory et OpenLDAP. D'autres types de serveurs LDAP peuvent fonctionner, mais

Amazon EMR ne prend pas officiellement en charge les autres types de serveurs. Pour plus d'informations, consultez Composants LDAP pour Amazon EMR.

`ActiveDirectoryConfigurations`

Sous-bloc obligatoire pour les configurations de sécurité utilisant le type de serveur Active Directory.

`ADDomain`

Option de console : Domaine Active Directory

Le nom de domaine utilisé pour créer le nom d'utilisateur principal (UPN) pour l'authentification des utilisateurs avec des configurations de sécurité utilisant le type de serveur Active Directory.

Considérations relatives aux configurations de sécurité avec LDAP et Amazon EMR

- Pour créer une configuration de sécurité avec l'intégration d'Amazon EMR LDAP, vous devez utiliser le chiffrement en transit. Pour plus d'informations sur le chiffrement en transit, consultez Chiffrer les données au repos et en transit.
- Vous ne pouvez pas définir la configuration Kerberos dans la même configuration de sécurité. Amazon EMR fournit un KDC dédié au KDC automatiquement et gère le mot de passe administrateur de ce KDC. Les utilisateurs ne peuvent pas accéder à ce mot de passe administrateur.
- Vous ne pouvez pas définir de rôles d'exécution IAM AWS Lake Formation dans la même configuration de sécurité.
- Le `LDAPServerURL` doit avoir le protocole `ldaps://` dans sa valeur.
- Le `LDAPAccessFilter` ne peut pas être vide.

Utilisation de LDAP avec l'intégration Apache Ranger pour Amazon EMR

Grâce à l'intégration LDAP pour Amazon EMR, vous pouvez poursuivre l'intégration avec Apache Ranger. Lorsque vous insérez des utilisateurs .your LDAP dans Ranger, vous pouvez ensuite associer ces utilisateurs à un serveur de règles Apache Ranger pour les intégrer à Amazon EMR et à d'autres applications. Pour ce faire, définissez le champ `RangerConfiguration` dans `AuthorizationConfiguration` dans la configuration de sécurité que vous utilisez avec votre cluster LDAP. Pour plus d'informations sur la configuration de la sécurité, consultez Création de la configuration de sécurité EMR.

Lorsque vous utilisez LDAP avec Amazon EMR, il n'est pas nécessaire de fournir une `KerberosConfiguration` avec l'intégration Amazon EMR pour Apache Ranger.

Lancement d'un cluster EMR qui s'authentifie auprès de LDAP

Procédez comme suit pour lancer un cluster EMR avec LDAP ou Active Directory.

1. Configuration de votre environnement :

   - Assurez-vous que les nœuds de votre cluster EMR peuvent communiquer avec Amazon S3 et. AWS Secrets Manager Pour plus d'informations sur la façon de modifier le rôle de votre profil d'instance EC2 afin de communiquer avec ces services, consultez Ajouter AWS Secrets Manager des autorisations au rôle d'instance Amazon EMR.
   - Si vous envisagez d'exécuter votre cluster EMR dans un sous-réseau privé, vous devez utiliser des points de terminaison AWS PrivateLink Amazon VPC ou utiliser la traduction d'adresses réseau (NAT) pour configurer le VPC afin qu'il communique avec S3 et Secrets Manager. Pour plus d'informations, consultez la section AWS PrivateLink et points de terminaison d'un VPC instances NAT dans le Guide de démarrage Amazon VPC.
   - Assurez-vous qu'il existe une connectivité réseau entre votre cluster EMR et le serveur LDAP. Vos clusters EMR doivent accéder à votre serveur LDAP via le réseau. Les nœuds principal, de noyau et de tâche du cluster communiquent avec le serveur LDAP pour synchroniser les données utilisateur. Si votre serveur LDAP s'exécute sur Amazon EC2, mettez à jour le groupe de sécurité EC2 pour accepter le trafic provenant du cluster EMR. Pour plus d'informations, consultez Ajouter AWS Secrets Manager des autorisations au rôle d'instance Amazon EMR.
2. Créez une configuration de sécurité Amazon EMR pour l'intégration LDAP. Pour plus d'informations, consultez Création de la configuration de sécurité Amazon EMR pour l'intégration LDAP.
3. Maintenant que vous êtes configuré, suivez les étapes décrites dans Lancement d'un cluster Amazon EMR pour lancer votre cluster avec les configurations suivantes :

   - Sélectionnez Amazon EMR version 6.12 ou supérieure. Nous vous recommandons d'utiliser la dernière version Amazon EMR.
   - Spécifiez ou sélectionnez uniquement les applications compatibles LDAP pour votre cluster. Pour obtenir la liste des applications prises en charge par LDAP avec Amazon EMR, consultez Prise en charge des applications et considérations relatives à LDAP pour Amazon EMR.
   - Appliquez la configuration de sécurité que vous avez créée à l'étape précédente.

Exemples d'utilisation de LDAP avec Amazon EMR

Une fois que vous avez configuré un cluster EMR utilisant l'intégration LDAP, vous pouvez fournir vos informations d'identification LDAP à n'importe quelle application prise en charge par le biais de son mécanisme d'authentification par nom d'utilisateur et mot de passe intégré. Cette page présente quelques exemples.

Utilisation de l'authentification LDAP avec Apache Hive

Example - Apache Hive

L'exemple de commande suivant démarre une session Apache Hive via HiveServer 2 et Beeline :

```
beeline -u "jdbc:hive2://$HOSTNAME:10000/default;ssl=true;sslTrustStore=
$TRUSTSTORE_PATH;trustStorePassword=$TRUSTSTORE_PASS"  -n LDAP_USERNAME -
p LDAP_PASSWORD
```

Utilisation de l'authentification LDAP avec Apache Livy

Example - Apache Livy

L'exemple de commande suivant démarre une session Livy via cURL. Remplacez *ENCODED-KEYPAIR* par une chaîne codée en Base64 pour `username:password`.

```
curl -X POST --data '{"proxyUser":"LDAP_USERNAME","kind": "pyspark"}' -H "Content-Type:
 application/json" -H "Authorization: Basic ENCODED-KEYPAIR" DNS_OF_PRIMARY_NODE:8998/
sessions
```

Utilisation de l'authentification LDAP avec Presto

Example - Presto

L'exemple de commande suivant démarre une session Presto via la CLI Presto :

```
presto-cli --user "LDAP_USERNAME" --password --catalog hive
```

Après avoir exécuté cette commande, saisissez le mot de passe LDAP à l'invite.

Utilisation de l'authentification LDAP avec Trino

Example - Turin

L'exemple de commande suivant démarre une session Trino via la CLI Trino :

```
trino-cli --user "LDAP_USERNAME" --password --catalog hive
```

Après avoir exécuté cette commande, saisissez le mot de passe LDAP à l'invite.

Utilisation de l'authentification LDAP avec Hue

Vous pouvez accéder à l'interface utilisateur de Hue via un tunnel SSH que vous créez sur le cluster, ou vous pouvez configurer un serveur proxy pour diffuser publiquement la connexion à Hue. Étant donné que Hue ne fonctionne pas en mode HTTPS par défaut, nous vous recommandons d'utiliser une couche de chiffrement supplémentaire pour garantir que les communications entre les clients et l'interface utilisateur de Hue sont chiffrées avec HTTPS. Cela réduit le risque que vous exposiez accidentellement les informations d'identification de l'utilisateur en texte brut.

Pour utiliser l'interface utilisateur Hue, ouvrez l'interface utilisateur Hue dans votre navigateur et entrez votre mot de passe LDAP pour vous connecter. Si les informations d'identification sont correctes, Hue vous connecte et utilise votre identité pour vous authentifier auprès de toutes les applications prises en charge.

Utilisation de SSH pour l'authentification par mot de passe et de tickets Kerberos pour d'autres applications

> **Important**
>
> Nous vous déconseillons d'utiliser l'authentification par mot de passe pour vous connecter à un cluster EMR.

Vous pouvez utiliser vos informations d'identification LDAP pour vous connecter en SSH à un cluster EMR. Pour ce faire, définissez la configuration `EnableSSHLogin` comme `true` dans la configuration de sécurité Amazon EMR que vous utilisez pour démarrer le cluster. Ensuite, utilisez la commande suivante pour vous connecter au cluster une fois qu'il a été lancé :

```
ssh username@EMR_PRIMARY_DNS_NAME
```

Après avoir exécuté cette commande, saisissez le mot de passe LDAP à l'invite.

Amazon EMR inclut un script intégré au cluster qui permet aux utilisateurs de générer un fichier keytab Kerberos et un ticket à utiliser avec les applications prises en charge qui n'acceptent pas

directement les informations d'identification LDAP. Certaines de ces applications incluent `spark-submit` Spark SQL et PySpark.

Exécutez `ldap-kinit` et suivez les instructions. Si l'authentification réussit, le fichier keytab Kerberos apparaît dans votre répertoire personnel avec un ticket Kerberos valide. Utilisez le ticket Kerberos pour exécuter des applications comme vous le feriez dans n'importe quel environnement activé pour Kerberos.

Intégrez Amazon EMR à AWS IAM Identity Center

Avec les versions 6.15.0 et supérieures d'Amazon EMR, vous pouvez utiliser les identités de pour vous authentifier auprès AWS IAM Identity Center d'un cluster Amazon EMR. Les pages suivantes fournissent une vue d’ensemble conceptuelle et présentent les prérequis et les étapes à suivre pour lancer un cluster EMR avec l’intégration d’Identity Center.

Rubriques

- Présentation
- Fonctionnalités et avantages
- Commencer à AWS IAM Identity Center intégrer Amazon EMR
- Considérations et limitations relatives à l’intégration d’Amazon EMR à Identity Center

Présentation

La propagation fiable des identités via IAM Identity Center peut vous aider à créer ou à connecter en toute sécurité les identités de vos employés, et à gérer de manière centralisée leur accès sur l'ensemble AWS des comptes et des applications. Grâce à cette fonctionnalité, un utilisateur peut se connecter à l'application qui utilise la propagation d'identité sécurisée, et cette application peut transmettre l'identité de l'utilisateur dans les demandes d'accès aux données dans des AWS services qui utilisent également la propagation d'identité sécurisée. L’accès étant géré en fonction de l’identité de l’utilisateur, les utilisateurs n’ont pas besoin d’utiliser les informations d’identification utilisateur locales de la base de données ni d’assumer un rôle IAM pour accéder aux données.

Identity Center est l'approche recommandée pour l'authentification et l'autorisation du personnel AWS pour les organisations de toutes tailles et de tous types. Avec Identity Center, vous pouvez créer et gérer les identités des utilisateurs dans AWS ou connecter votre source d'identité existante, notamment Microsoft Active Directory, Okta, Ping Identity JumpCloud, Google Workspace et Microsoft Entra ID (anciennement Azure AD).

Pour plus d'informations, voir Qu'est-ce que c'est AWS IAM Identity Center ? et Propagation fiable de l'identité entre les applications dans le Guide de AWS IAM Identity Center l'utilisateur.

Fonctionnalités et avantages

L’intégration d’Amazon EMR à IAM Identity Center offre les avantages suivants :

- Amazon EMR fournit des informations d’identification pour relayer l’identité Identity Center vers un cluster EMR.
- Amazon EMR configure toutes les applications prises en charge pour qu’elles s’authentifient avec les informations d’identification du cluster.
- Amazon EMR configure et maintient la sécurité des applications prises en charge avec le protocole Kerberos, et aucune commande ni aucun script ne vous sont demandés.
- Vous avez la possibilité d’appliquer l’autorisation au niveau du préfixe Amazon S3 avec les identités Identity Center aux préfixes S3 gérés par S3 Access Grants.
- Possibilité d'appliquer l'autorisation au niveau des tables avec les identités Identity Center sur les tables AWS Lake Formation AWS Glue gérées.

Commencer à AWS IAM Identity Center intégrer Amazon EMR

Cette section vous aide à configurer Amazon EMR pour l'intégrer à. AWS IAM Identity Center

Rubriques

- Création d’une instance d’Identity Center
- Création d’un rôle IAM pour Identity Center
- Création d’une configuration de sécurité compatible avec Identity Center
- Création et lancement d’un cluster compatible avec Identity Center
- Configuration de Lake Formation pour un cluster EMR compatible avec IAM Identity Center
- Utilisation de S3 Access Grants sur un cluster EMR compatible avec IAM Identity Center

Création d’une instance d’Identity Center

Si vous n’en avez pas encore, créez une instance d’Identity Center dans la Région AWS où vous souhaitez lancer votre cluster EMR. Une instance d’Identity Center ne peut exister que dans une seule région pour un Compte AWS.

Utilisez la AWS CLI commande suivante pour créer une nouvelle instance nommée *MyInstance* :

```
aws sso-admin create-instance --name MyInstance
```

Création d’un rôle IAM pour Identity Center

Pour intégrer Amazon EMR à AWS IAM Identity Center, créez un rôle IAM qui s'authentifie auprès d'Identity Center à partir du cluster EMR. En arrière-plan, Amazon EMR utilise des informations d’identification SigV4 pour transmettre l’identité Identity Center à des services en aval comme AWS Lake Formation. Votre rôle doit également disposer des autorisations correspondantes pour invoquer les services en aval.

Lorsque vous créez le rôle, utilisez la politique d’autorisations suivante :

```
{
  "Statement": [
    {
      "Sid": "IdCPermissions",
      "Effect": "Allow",
      "Action": [
        "sso-oauth:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "GlueandLakePermissions",
      "Effect": "Allow",
      "Action": [
        "glue:*",
        "lakeformation:GetDataAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AccessGrantsPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix"
      ],
      "Resource": "*"
    }
```

```
  ]
}
```

La politique de confiance de ce rôle permet au rôle InstanceProfile d'assumer le rôle.

```
{
   "Sid": "AssumeRole",
   "Effect": "Allow",
   "Principal": {
        "AWS": "arn:aws:iam::12345678912:role/EMR_EC2_DefaultRole"
    },
    "Action": [
               "sts:AssumeRole",
               "sts:SetContext"
    ]
 }
```

Création d'une configuration de sécurité compatible avec Identity Center

Pour lancer un cluster EMR avec l'intégration d'IAM Identity Center, utilisez l'exemple de commande suivant pour créer une configuration de sécurité Amazon EMR pour laquelle Identity Center est activé. Chaque configuration est détaillée ci-dessous.

```
aws emr create-security-configuration --name "IdentityCenterConfiguration-with-lf-
accessgrants" --region "us-west-2" --security-configuration '{
 "AuthenticationConfiguration":{
  "IdentityCenterConfiguration":{
   "EnableIdentityCenter":true,
   "IdentityCenterApplicationAssigmentRequired":false,
   "IdentityCenterInstanceARN": "arn:aws:sso:::instance/ssoins-123xxxxxxxxx789",
   "IAMRoleForEMRIdentityCenterApplicationARN": "arn:aws:iam::123456789012:role/tip-
role"
     }
 },
 "AuthorizationConfiguration": {
  "LakeFormationConfiguration": {
   "EnableLakeFormation": true
  }
 },
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": false,
```

```
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://my-bucket/cert/my-certs.zip"
   }
  }
 }
}'
```

- **`EnableIdentityCenter`** (obligatoire) : active l'intégration d'Identity Center
- **`IdentityCenterApplicationARN`** (obligatoire) : ARN de l'instance Identity Center
- **`IAMRoleForEMRIdentityCenterApplicationARN`** (obligatoire) : rôle IAM qui récupère les jetons Identity Center auprès du cluster
- **`IdentityCenterApplicationAssignmentRequired`** (booléen) : détermine si une affectation est requise pour l'utilisation de l'application Identity Center La valeur par défaut est `true`.
- **`AuthorizationConfiguration`/`LakeFormationConfiguration`**— Configurez éventuellement l'autorisation :
  - **`EnableLakeFormation`** : active l'autorisation Lake Formation sur le cluster

Spécifiez `EncryptionConfiguration` et `IntransitEncryptionConfiguration` pour activer l'intégration d'Identity Center à Amazon EMR.

Création et lancement d'un cluster compatible avec Identity Center

Maintenant que vous avez configuré le rôle IAM qui s'authentifie auprès d'Identity Center et que vous avez créé une configuration de sécurité Amazon EMR pour laquelle Identity Center est activé, vous pouvez créer et lancer votre cluster basé sur l'identité. Pour savoir comment lancer votre cluster avec la configuration de sécurité requise, consultez la rubrique Spécification d'une configuration de sécurité pour un cluster.

Reportez-vous également à la section suivante si vous souhaitez utiliser votre cluster compatible avec Identity Center avec d'autres options de sécurité prises en charge par Amazon EMR :

- Utilisation de S3 Access Grants sur un cluster EMR compatible avec IAM Identity Center
- Configuration de Lake Formation pour un cluster EMR compatible avec IAM Identity Center

Configuration de Lake Formation pour un cluster EMR compatible avec IAM Identity Center

Vous pouvez l'intégrer AWS Lake Formationà votre AWS IAM Identity Center cluster EMR activé.

Tout d'abord, assurez-vous qu'une instance Identity Center est configurée dans la même région que votre cluster. Pour plus d'informations, consultez Création d'une instance d'Identity Center. Pour afficher l'ARN de l'instance, accédez aux détails de l'instance dans la console IAM Identity Center ou utilisez la commande suivante pour afficher les détails de toutes vos instances depuis la CLI :

```
aws sso-admin list-instances
```

Utilisez ensuite l'ARN et l'ID de votre AWS compte avec la commande suivante pour configurer Lake Formation afin qu'il soit compatible avec IAM Identity Center :

```
aws lakeformation create-lake-formation-identity-center-configuration --cli-input-json
 file://create-lake-fromation-idc-config.json
json input:
{
    "CatalogId": "account-id/org-account-id",
    "InstanceArn": "identity-center-instance-arn"
}
```

À présent, appelez `put-data-lake-settings` et activez `AllowFullTableExternalDataAccess` avec Lake Formation :

```
aws lakeformation put-data-lake-settings --cli-input-json file://put-data-lake-
settings.json
json input:
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier": "admin-ARN"
            }
        ],
        "CreateDatabaseDefaultPermissions": [...],
        "CreateTableDefaultPermissions": [...],
        "AllowExternalDataFiltering": true,
        "AllowFullTableExternalDataAccess": true
```

```
    }
}
```

Enfin, accordez des autorisations de table complètes à l'ARN d'identité pour l'utilisateur qui accède au cluster EMR. L'ARN contient l'ID utilisateur d'Identity Center. Accédez à Identity Center dans la console, sélectionnez Utilisateurs, puis sélectionnez l'utilisateur pour afficher ses paramètres dans Informations générales.

Copiez l'ID utilisateur et collez-le dans l'ARN suivant pour *user-id* :

```
arn:aws:identitystore:::user/user-id
```

> **Note**
>
> Les requêtes sur le cluster EMR ne fonctionnent que si l'identité Identity Center IAM dispose d'un accès complet à la table protégée de Lake Formation. Dans le cas contraire, la requête échouera.

Utilisez la commande suivante pour accorder à l'utilisateur l'accès complet aux tables :

```
aws lakeformation grant-permissions --cli-input-json file://grantpermissions.json
json input:
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:identitystore:::user/user-id"
    },
    "Resource": {
        "Table": {
            "DatabaseName": "tip_db",
            "Name": "tip_table"
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": [
        "ALL"
    ]
}
```

Utilisation de S3 Access Grants sur un cluster EMR compatible avec IAM Identity Center

Vous pouvez intégrer S3 Access Grants à votre AWS IAM Identity Center cluster EMR activé.

Utilisez S3 Access Grants pour autoriser l'accès à vos jeux de données résidant sur des clusters utilisant Identity Center. Créez des octrois pour augmenter les autorisations que vous définissez pour les utilisateurs IAM, les groupes, les rôles ou pour un annuaire d'entreprise. Pour plus d'informations, voir la rubrique Utilisation de S3 Access Grants avec Amazon EMR.

Rubriques

- Création d'une instance et d'un emplacement S3 Access Grants
- Créez des octrois pour les identités Identity Center

Création d'une instance et d'un emplacement S3 Access Grants

Si vous n'en avez pas déjà une, créez une instance S3 Access Grants dans la Région AWS où vous souhaitez lancer votre cluster EMR.

Utilisez la AWS CLI commande suivante pour créer une nouvelle instance nommée *MyInstance* :

```
aws s3control-access-grants create-access-grants-instance \
--account-id 12345678912 \
--identity-center-arn "identity-center-instance-arn" \
```

Créez ensuite un emplacement S3 Access Grants, en remplaçant les valeurs en rouge par les vôtres :

```
aws s3control-access-grants create-access-grants-location \
--account-id 12345678912 \
--location-scope s3:// \
--iam-role-arn "access-grant-role-arn" \
--region aa-example-1
```

> **Note**
>
> Utilisez l'ARN `accessGrantRole` comme valeur pour le paramètre `iam-role-arn`.

Créez des octrois pour les identités Identity Center

Enfin, créez les octrois pour les identités ayant accès à votre cluster :

```
aws s3control-access-grants create-access-grant \
--account-id 12345678912 \
--access-grants-location-id "default" \
--access-grants-location-configuration S3SubPrefix="s3-bucket-prefix"
--permission READ \
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier="your-identity-center-user-id"
```

Exemple de sortie :

```
{
"CreatedAt": "2023-09-21T23:47:24.870000+00:00",
"AccessGrantId": "1234-12345-1234-1234567",
"AccessGrantArn": "arn:aws:s3:aa-example-1-1:123456789012:access-grants/default/grant/
xxxx1234-1234-5678-1234-1234567890",
"Grantee": {
"GranteeType": "DIRECTORY_USER",
"GranteeIdentifier": "5678-56789-5678-567890"
},
"AccessGrantsLocationId": "default",
"AccessGrantsLocationConfiguration": {
"S3SubPrefix": "myprefix/*"
},
"Permission": "READ",
"GrantScope": "s3://myprefix/*"
}
```

Considérations et limitations relatives à l'intégration d'Amazon EMR à Identity Center

Tenez compte des points suivants lorsque vous utilisez IAM Identity Center avec Amazon EMR :

- La propagation d'identité approuvée via Identity Center est prise en charge sur Amazon EMR 6.15.0 et versions ultérieures, et uniquement avec Apache Spark.
- Pour activer les clusters EMR avec propagation d'identité sécurisée, vous devez utiliser le AWS CLI pour créer une configuration de sécurité dans laquelle la propagation d'identité sécurisée

est activée, et utiliser cette configuration de sécurité lorsque vous lancez votre cluster. Pour plus d'informations, consultez Création d'une configuration de sécurité compatible avec Identity Center.

- Les clusters EMR qui utilisent la propagation d'identité approuvée ne peuvent invoquer que les services qui utilisent également la propagation d'identité approuvée.
- Seul le contrôle d'accès au niveau de la table basé sur AWS Lake Formation est disponible pour les clusters EMR qui utilisent une propagation d'identité sécurisée.
- Pour les clusters EMR qui utilisent la propagation d'identité approuvée, les opérations qui prennent en charge le contrôle d'accès basé sur Lake Formation avec Apache Spark incluent `SELECT`, `ALTER TABLE` et `DROP TABLE`.
- Dans le cas des clusters EMR qui utilisent la propagation d'identité approuvée, les contrôles d'accès basés sur Lake Formation qui ne sont pas pris en charge par Apache Spark incluent les instructions `INSERT`.
- La propagation d'identités fiables avec Amazon EMR est prise en charge dans les pays suivants : Régions AWS
  - `ap-east-1` – Asie-Pacifique (Hong Kong)
  - `ap-northeast-1` – Asie-Pacifique (Tokyo)
  - `ap-northeast-2` – Asie-Pacifique (Séoul)
  - `ap-south-1` – Asie-Pacifique (Mumbai)
  - `ap-southeast-1` – Asie-Pacifique (Singapour)
  - `ap-southeast-2` – Asie-Pacifique (Sydney)
  - `ca-central-1` – Canada (Centre)
  - `eu-central-1` – Europe (Francfort)
  - `eu-north-1` – Europe (Stockholm)
  - `eu-west-1` – Europe (Irlande)
  - `eu-west-2` – Europe (Londres)
  - `eu-west-3` – Europe (Paris)
  - `me-south-1` – Moyen-Orient (Bahreïn)
  - `sa-east-1` – Amérique du Sud (São Paulo)
  - `us-east-1` – USA Est (Virginie du Nord)
  - `us-east-2` – USA Est (Ohio)
  - `us-west-1` – USA Ouest (Californie du Nord)
  - `us-west-2` – USA Ouest (Oregon)

Intégrez Amazon EMR à AWS Lake Formation

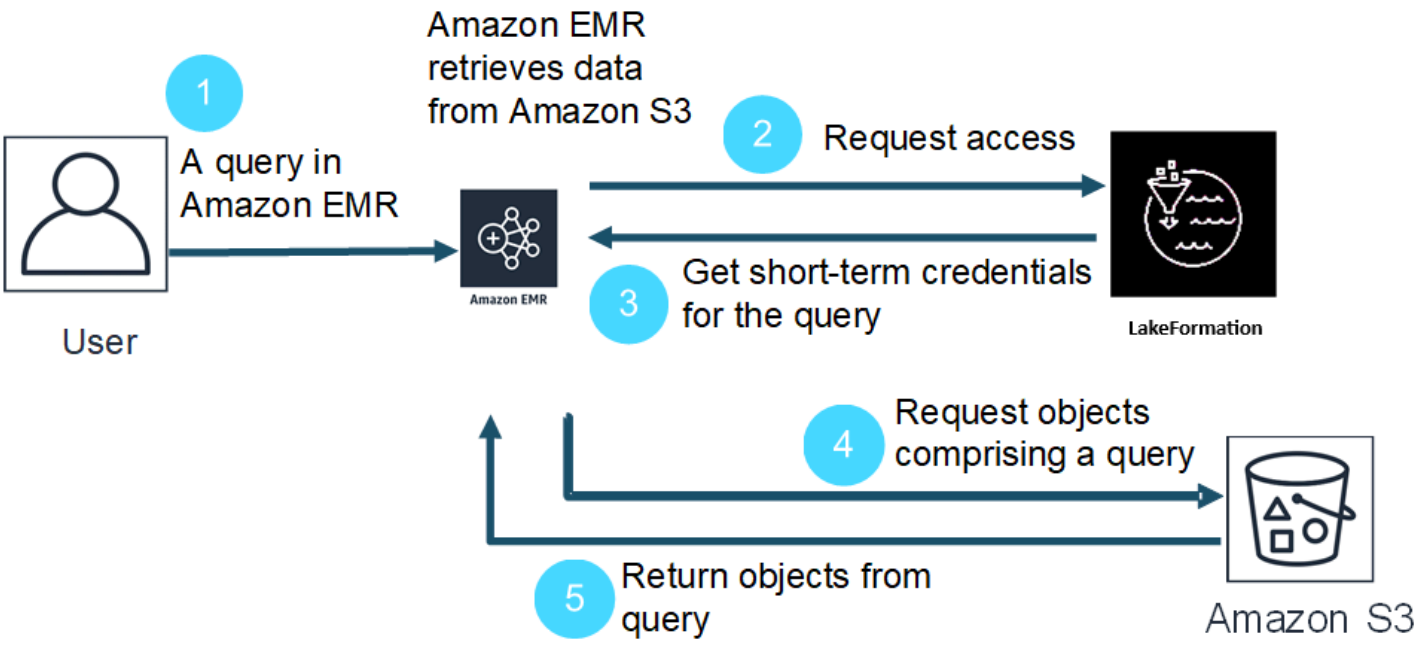
AWS Lake Formation est un service géré qui vous permet de découvrir, de cataloguer, de nettoyer et de sécuriser les données dans un lac de données Amazon Simple Storage Service (S3). Lake Formation fournit un accès détaillé au niveau des colonnes aux bases de données et aux tables du Glue AWS Data Catalog. Pour plus d'informations, consultez Qu'est-ce que AWS Lake Formation ?

Avec Amazon EMR version 6.7.0 et versions ultérieures, vous pouvez appliquer un contrôle d'accès basé sur Lake Formation aux tâches Spark, Hive et Presto que vous soumettez aux clusters Amazon EMR. Pour intégrer Lake Formation, vous devez créer un cluster EMR doté d'un rôle d'exécution. Un rôle d'exécution est un rôle AWS Identity and Access Management (IAM) que vous associez à des tâches ou à des requêtes Amazon EMR. Amazon EMR utilise ensuite ce rôle pour accéder AWS aux ressources. Pour plus d'informations, consultez Rôles d'exécution pour les étapes Amazon EMR.

Comment Amazon EMR fonctionne avec Lake Formation

Après avoir intégré Amazon EMR à Lake Formation, vous pouvez exécuter des requêtes vers les clusters Amazon EMR à l'aide de l'`StepAPI` ou de Studio. SageMaker Lake Formation fournit ensuite un accès aux données via des informations d'identification temporaires pour Amazon EMR. Ce processus est appelé distributeur d'informations d'identification. Pour plus d'informations, consultez Qu'est-ce que AWS Lake Formation ?

Voici un aperçu de haut niveau de la manière dont Amazon EMR accède aux données protégées par les politiques de sécurité de Lake Formation.



1. Un utilisateur soumet une requête Amazon EMR pour des données dans Lake Formation.
2. Amazon EMR demande des informations d'identification temporaires à Lake Formation pour permettre à l'utilisateur d'accéder aux données.
3. Lake Formation renvoie des informations d'identification temporaires.
4. Amazon EMR envoie la demande de requête pour récupérer les données d'Amazon S3.
5. Amazon EMR reçoit les données d'Amazon S3, les filtre et renvoie les résultats en fonction des autorisations utilisateur définies par l'utilisateur dans Lake Formation.

Pour plus d'informations sur l'ajout d'utilisateurs et de groupes aux politiques de Lake Formation, consultez la section Octroi d'autorisations au catalogue de données.

Prérequis

Vous devez remplir les conditions suivantes avant d'intégrer Amazon EMR et Lake Formation :

- Activez l'autorisation des rôles d'exécution sur votre cluster Amazon EMR.
- Utilisez le catalogue de données AWS Glue comme magasin de métadonnées.
- Définissez et gérez les autorisations dans Lake Formation pour accéder aux bases de données, aux tables et aux colonnes de AWS Glue Data Catalog. Pour plus d'informations, consultez Qu'est-ce que AWS Lake Formation ?

Rubriques

- Activation de Lake Formation avec Amazon EMR
- Apache Hudi et Lake Formation
- Apache Iceberg et Lake Formation
- Delta Lake et Lake Formation
- Considérations relatives à Amazon EMR avec Lake Formation

Activation de Lake Formation avec Amazon EMR

Avec Amazon EMR 6.15.0 et versions ultérieures, lorsque vous exécutez des tâches Spark sur Amazon EMR sur des clusters EC2 qui accèdent aux données du AWS Glue Data Catalog, vous pouvez appliquer des autorisations au niveau des tables, des lignes, des colonnes et des cellules AWS Lake Formation aux tables basées sur Hudi, Iceberg ou Delta Lake.

Dans cette section, nous expliquons comment créer une configuration de sécurité et configurer Lake Formation pour qu'il fonctionne avec Amazon EMR. Nous expliquons également comment lancer un cluster avec la configuration de sécurité que vous avez créée pour Lake Formation.

Étape 1 : configurer un rôle d'exécution pour votre cluster EMR

Pour utiliser un rôle d'exécution pour votre cluster EMR, vous devez créer une configuration de sécurité. Avec une configuration de sécurité, vous pouvez appliquer des options de sécurité, d'autorisation et d'authentification cohérentes sur l'ensemble de vos clusters.

1. Créez un fichier appelé `lf-runtime-roles-sec-cfg.json` avec la configuration de sécurité suivante.

```
{
    "AuthorizationConfiguration": {
        "IAMConfiguration": {
            "EnableApplicationScopedIAMRole": true,
            "ApplicationScopedIAMRoleConfiguration": {
                "PropagateSourceIdentity": true
            }
        },
        "LakeFormationConfiguration": {
            "AuthorizedSessionTagValue": "Amazon EMR"
        }
```

```
    },
    "EncryptionConfiguration": {
            "EnableInTransitEncryption": true,
            "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {<certificate-configuration>}
        }
    }
}
```

2. Ensuite, pour vous assurer que le tag de session peut autoriser Lake Formation, définissez la propriété `LakeFormationConfiguration/AuthorizedSessionTagValue` sur `Amazon EMR`.
3. Utilisez la commande suivante pour créer la configuration de sécurité Amazon EMR.

   ```
   aws emr create-security-configuration \
   --name 'iamconfig-with-iam-lf' \
   --security-configuration file://lf-runtime-roles-sec-cfg.json
   ```

   Vous pouvez également utiliser la console Amazon EMR pour créer une configuration de sécurité avec des paramètres personnalisés.

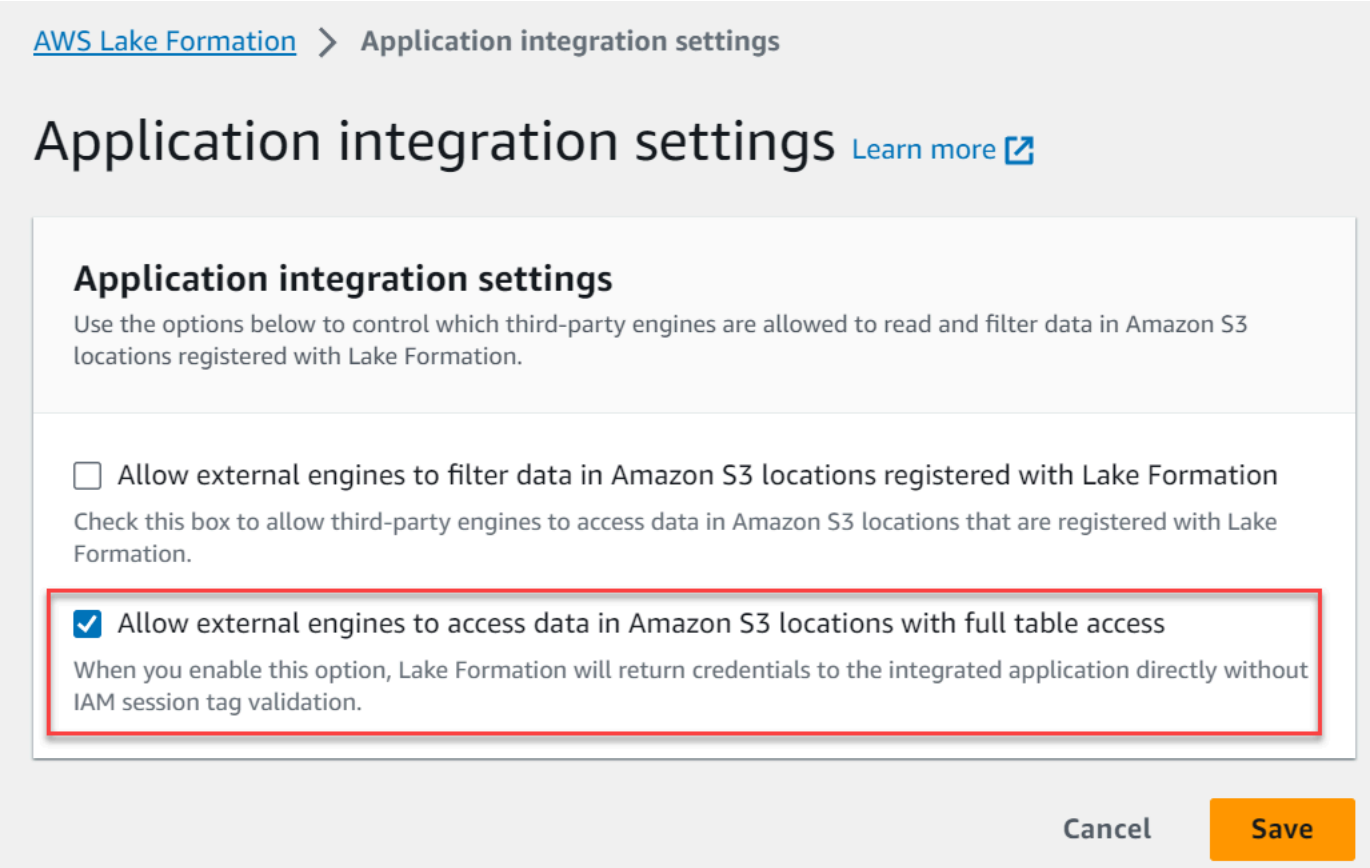
Étape 2 : lancer un cluster Amazon EMR

Vous êtes maintenant prêt à lancer un cluster EMR avec la configuration de sécurité que vous avez créée à l'étape précédente. Pour plus d'informations sur les configurations de sécurité, consultez Utilisation de configurations de sécurité pour configurer la sécurité du cluster et Rôles d'exécution pour les étapes Amazon EMR.

Étape 3a : configurer les autorisations au niveau des tables basées sur Lake Formation avec les rôles d’exécution Amazon EMR

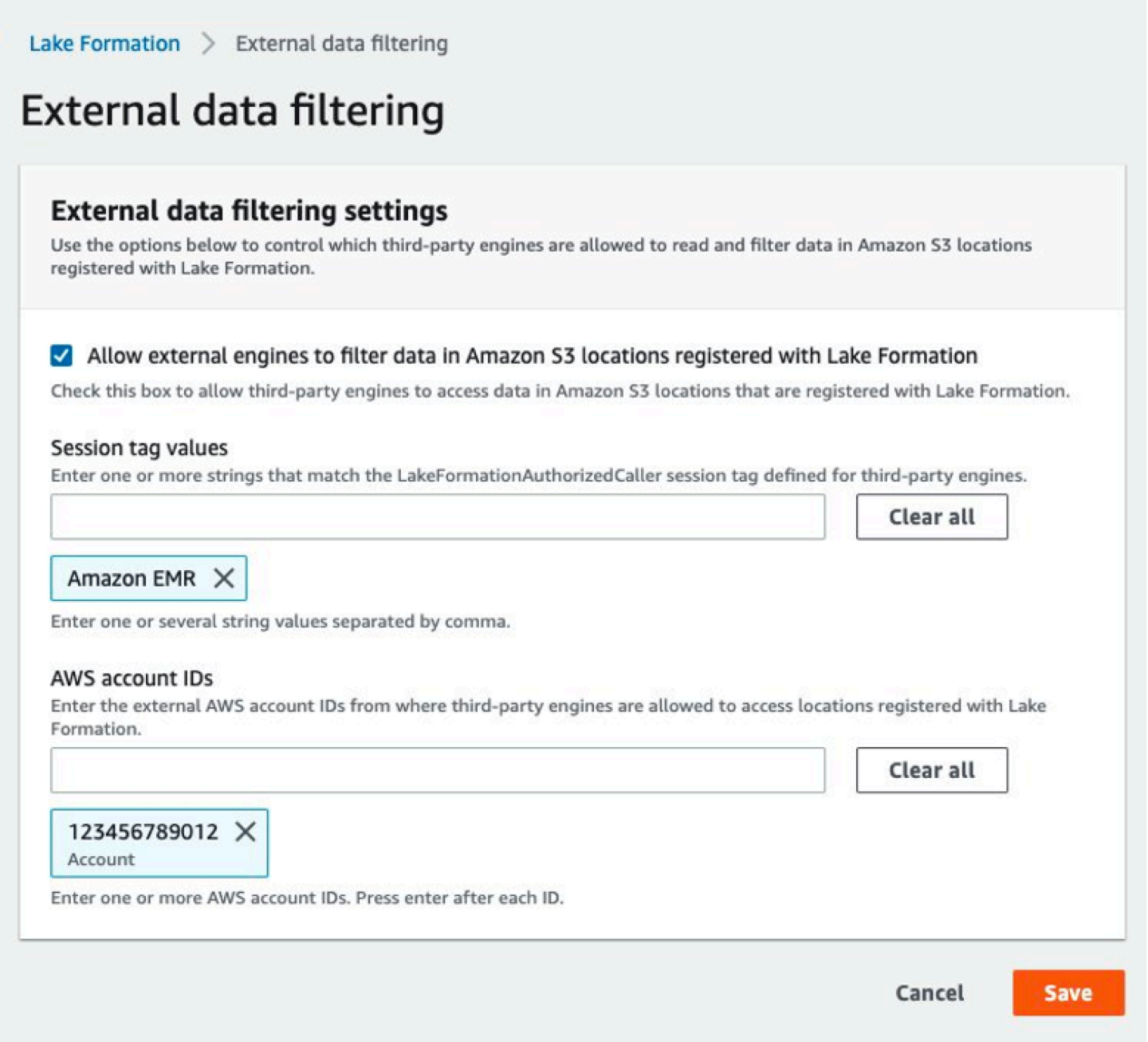
Si vous n’avez pas besoin d’un contrôle d’accès précis au niveau des colonnes, des lignes ou des cellules, vous pouvez configurer des autorisations au niveau des tables avec le catalogue de données Glue. Pour activer l'accès au niveau de la table, accédez à la AWS Lake Formation console et sélectionnez l'option Paramètres d'intégration des applications dans la section Administration de la barre latérale. Activez ensuite l’option suivante et choisissez Enregistrer :

Autoriser les moteurs externes à accéder aux données dans les emplacements Amazon S3 avec un accès complet aux tables



Étape 3b : configurer les autorisations au niveau des colonnes, des lignes ou des cellules basées sur Lake Formation avec les rôles d'exécution Amazon EMR

Pour appliquer des autorisations au niveau des tables et des colonnes avec Lake Formation, l'administrateur du lac de données de Lake Formation doit définir `Amazon EMR` comme valeur pour la configuration des balises de session, `AuthorizedSessionTagValue`. Lake Formation utilise cette balise de session pour autoriser les appelants et fournir l'accès au lac de données. Vous pouvez définir cette balise de session dans la section Filtrage des données externes de la console Lake Formation. Remplacez *123456789012* par votre propre identifiant Compte AWS .



Étape 4 : Configuration des subventions AWS Glue et Lake Formation pour les rôles d'exécution Amazon EMR

Pour poursuivre la configuration du contrôle d'accès basé sur Lake Formation avec les rôles d'exécution Amazon EMR, vous devez configurer les subventions AWS Glue et Lake Formation pour les rôles d'exécution Amazon EMR. Pour permettre à vos rôles d'exécution IAM d'interagir avec Lake Formation, accordez-leur l'accès avec `lakeformation:GetDataAccess` et`glue:Get*`.

Les autorisations de Lake Formation contrôlent l'accès aux ressources du catalogue de données AWS Glue, aux sites Amazon S3 et aux données sous-jacentes de ces sites. Les autorisations IAM contrôlent l'accès aux API et aux ressources de Lake Formation et AWS Glue. Bien que vous ayez l'autorisation Lake Formation d'accéder à une table du catalogue de données (SELECT), votre opération échoue si vous ne disposez pas de l'autorisation IAM sur l'API `glue:Get*`. Pour plus de détails sur le contrôle d'accès à Lake Formation, consultez Présentation du contrôle d'accès à Lake Formation.

1. Créez le fichier `emr-runtime-roles-lake-formation-policy.json` avec le contenu suivant.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "LakeFormationManagedAccess",
        "Effect": "Allow",
        "Action": [
            "lakeformation:GetDataAccess",
            "glue:Get*",
            "glue:Create*",
            "glue:Update*"
        ],
        "Resource": "*"
    }
}
```

2. Créez la politique IAM correspondante.

```
aws iam create-policy \
--policy-name emr-runtime-roles-lake-formation-policy \
--policy-document file://emr-runtime-roles-lake-formation-policy.json
```

3. Pour attribuer cette politique à vos rôles d'exécution IAM, suivez les étapes de la section Gestion des autorisations AWS Lake Formation.

Vous pouvez désormais utiliser les rôles d'exécution et Lake Formation pour appliquer des autorisations au niveau des tables et des colonnes. Vous pouvez également utiliser une identité source pour contrôler les actions et surveiller les opérations avec AWS CloudTrail. Pour un end-to-end exemple détaillé, consultez Présentation des rôles d'exécution pour les étapes Amazon EMR.

Apache Hudi et Lake Formation

Les versions 6.15.0 et supérieures d'Amazon EMR incluent la prise en charge d'un contrôle d'accès précis basé sur Apache Hudi lorsque vous lisez et écrivez des données AWS Lake Formation avec Spark SQL. Amazon EMR prend en charge le contrôle d'accès au niveau des tables, des lignes, des colonnes et des cellules avec Apache Hudi. Grâce à cette fonctionnalité, vous pouvez exécuter des requêtes instantanées sur copy-on-write des tables pour demander le dernier instantané de la table à un instant de validation ou de compactage donné.

Actuellement, un cluster Amazon EMR compatible avec Lake Formation doit récupérer la colonne de temps de validation de Hudi pour effectuer des requêtes incrémentielles et des requêtes de voyage dans le temps. Il ne prend pas en charge la `timestamp as of` syntaxe et la `Spark.read()` fonction de Spark. La syntaxe correcte est`select * from table where _hoodie_commit_time <= point_in_time`. Pour plus d'informations, voir Requêtes ponctuelles sur le voyage dans le temps sur la table Hudi.

La matrice de support suivante répertorie certaines fonctionnalités essentielles d'Apache Hudi avec Lake Formation :

	Copie sur écriture	fusion sur lecture
Requêtes d'instantanés – Spark SQL	✓	✓
Requêtes optimisées en lecture – Spark SQL	✓	✓
Requêtes progressives	✓	✓
Requêtes Time Travel	✓	✓
Tables de métadonnées	✓	✓
Commandes DML **`INSERT`**	✓	✓
Commandes DML		
Requêtes de source de données Spark		
Écritures de source de données Spark		

Interrogation des tables Hudi

Cette section explique comment exécuter les requêtes prises en charge décrites ci-dessus sur un cluster compatible avec Lake Formation. La table doit être une table de catalogue enregistrée.

1. Pour démarrer le shell Spark, utilisez les commandes suivantes :

```
spark-sql
--jars /usr/lib/hudi/hudi-spark-bundle.jar \
--conf spark.serializer=org.apache.spark.serializer.KryoSerializer \
--conf
spark.sql.catalog.spark_catalog=org.apache.spark.sql.hudi.catalog.HoodieCatalog \
--conf
spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension,com.amazonaws.em
 \
--conf spark.sql.catalog.spark_catalog.lf.managed=true
```

   Si vous souhaitez que Lake Formation utilise un serveur d'enregistrement pour gérer votre catalogue Spark, définissez `spark.sql.catalog.<managed_catalog_name>.lf.managed` ce paramètre sur true.

2. Pour demander le dernier instantané des copy-on-write tables, utilisez les commandes suivantes.

```
SELECT * FROM my_hudi_cow_table
```

```
spark.read.table("my_hudi_cow_table")
```

3. Pour interroger les dernières données compactées des tables `MOR`, vous pouvez interroger la table optimisée en lecture qui est suffixée par `_ro` :

```
SELECT * FROM my_hudi_mor_table_ro
```

```
spark.read.table("my_hudi_mor_table_ro")
```

> **Note**
>
> Les performances des lectures sur les clusters de Lake Formation peuvent être plus lentes en raison d'optimisations non prises en charge. Ces fonctionnalités incluent la liste des fichiers

basée sur les métadonnées Hudi et le saut de données. Nous vous recommandons de tester les performances de votre application pour vous assurer qu'elle répond à vos exigences.

Apache Iceberg et Lake Formation

Les versions 6.15.0 et supérieures d'Amazon EMR incluent la prise en charge d'un contrôle d'accès précis basé sur Apache Iceberg lorsque vous lisez et écrivez des données AWS Lake Formation avec Spark SQL. Amazon EMR prend en charge le contrôle d'accès au niveau des tables, des lignes, des colonnes et des cellules avec Apache Iceberg. Grâce à cette fonctionnalité, vous pouvez exécuter des requêtes instantanées sur copy-on-write des tables pour demander le dernier instantané de la table à un instant de validation ou de compactage donné.

Si vous souhaitez utiliser le format Iceberg, définissez les configurations suivantes. Remplacez *DB_LOCATION* par l'emplacement Amazon S3 de vos tables Iceberg, et remplacez les espaces réservés à la région et à l'ID de compte par vos propres valeurs.

```
spark-sql \
--conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions,com.ama

--conf spark.sql.catalog.iceberg_catalog=org.apache.iceberg.spark.SparkCatalog
--conf spark.sql.catalog.iceberg_catalog.warehouse=s3://DB_LOCATION
--conf spark.sql.catalog.iceberg_catalog.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog
--conf spark.sql.catalog.iceberg_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO
--conf spark.sql.catalog.iceberg_catalog.glue.account-id=ACCOUNT_ID
--conf spark.sql.catalog.iceberg_catalog.glue.id=ACCOUNT_ID
--conf spark.sql.catalog.iceberg_catalog.client.assume-role.region=AWS_REGION
--conf spark.sql.secureCatalog=iceberg_catalog
```

Si vous souhaitez que Lake Formation utilise un serveur d'enregistrement pour gérer votre catalogue Spark, définissez `spark.sql.catalog.<managed_catalog_name>.lf.managed` ce paramètre sur true.

Veillez également à NE PAS transmettre les paramètres d'acceptation de rôle suivants :

```
--conf spark.sql.catalog.my_catalog.client.assume-role.region
--conf spark.sql.catalog.my_catalog.client.assume-role.arn
```

```
--conf spark.sql.catalog.my_catalog.client.assume-
role.tags.LakeFormationAuthorizedCaller
```

La matrice de prise en charge suivante répertorie certaines fonctionnalités essentielles d'Apache Iceberg avec Lake Formation :

	Copie sur écriture	fusion sur lecture
Requêtes d'instantanés – Spark SQL	✓	✓
Requêtes optimisées en lecture – Spark SQL	✓	✓
Requêtes progressives	✓	✓
Requêtes Time Travel	✓	✓
Tables de métadonnées	✓	✓
Commandes DML **`INSERT`**	✓	✓
Commandes DML		
Requêtes de source de données Spark		
Écritures de source de données Spark		

Delta Lake et Lake Formation

Les versions 6.15.0 et supérieures d'Amazon EMR incluent la prise en charge d'un contrôle d'accès précis basé sur Delta Lake lorsque vous lisez et écrivez des données AWS Lake Formation avec Spark SQL. Amazon EMR prend en charge le contrôle d'accès au niveau des tables, des lignes, des colonnes et des cellules avec Delta Lake. Grâce à cette fonctionnalité, vous pouvez exécuter des requêtes instantanées sur copy-on-write des tables pour demander le dernier instantané de la table à un instant de validation ou de compactage donné.

Pour utiliser Delta Lake with Lake Formation, exécutez la commande suivante.

```
spark-sql \
```

```
--conf
 spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension,com.amazonaws.emr.recordserver.co
 \
--conf spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog
 \
--conf spark.sql.catalog.spark_catalog.lf.managed=true
```

Si vous souhaitez que Lake Formation utilise un serveur d'enregistrement pour gérer votre catalogue Spark, définissez `spark.sql.catalog.<managed_catalog_name>.lf.managed` ce paramètre sur true.

La matrice de prise en charge suivante répertorie certaines fonctionnalités essentielles de Delta Lake avec Lake Formation :

	Copie sur écriture	fusion sur lecture
Requêtes d'instantanés – Spark SQL	✓	✓
Requêtes optimisées en lecture – Spark SQL	✓	✓
Requêtes progressives	Non pris en charge	Non pris en charge
Requêtes Time Travel	Non pris en charge	Non pris en charge
Tables de métadonnées	✓	✓
Commandes DML **`INSERT`**	✓	✓
Commandes DML		
Requêtes de source de données Spark		
Écritures de source de données Spark		

Création d'une table Delta Lake dans AWS Glue Data Catalog

Amazon EMR with Lake Formation ne prend pas en charge les commandes DDL ni la création de tables Delta. Procédez comme suit pour créer des tables dans le catalogue de données AWS Glue.

1. Utilisez l'exemple suivant pour créer une table Delta. Assurez-vous que votre emplacement S3 existe.

```
spark-sql \
--conf "spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension" \
--conf
 "spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog"

> CREATE DATABASE if not exists <DATABASE_NAME> LOCATION 's3://<S3_LOCATION>/
transactionaldata/native-delta/<DATABASE_NAME>/';
> CREATE TABLE <TABLE_NAME> (x INT, y STRING, z STRING) USING delta;
> INSERT INTO <TABLE_NAME> VALUES (1, 'a1', 'b1');
```

2. Pour consulter les détails de votre tableau, rendez-vous sur https://console.aws.amazon.com/glue/.
3. Dans le volet de navigation de gauche, développez le catalogue de données, choisissez Tables, puis choisissez la table que vous avez créée. Sous Schema, vous devriez voir que la table Delta que vous avez créée avec Spark stocke toutes les colonnes dans un type de `array<string>` données tel que AWS Glue.
4. Pour définir des filtres au niveau des colonnes et des cellules dans Lake Formation, supprimez la `col` colonne de votre schéma, puis ajoutez les colonnes figurant dans le schéma de votre table. Dans cet exemple, ajoutez les colonnes `xy`, `etz`.

Considérations relatives à Amazon EMR avec Lake Formation

Tenez compte des points suivants lorsque vous utilisez Amazon EMR avec. AWS Lake Formation

- Le contrôle d'accès au niveau des tables est disponible sur les clusters dotés de la version 6.13 d'Amazon EMR ou d'une version ultérieure.
- le contrôle d'accès précis au niveau des lignes, des colonnes et des cellules est disponible sur les clusters dotés de la version 6.15 d'Amazon EMR ou d'une version ultérieure.
- Les utilisateurs ayant accès à une table peuvent accéder à toutes les propriétés de cette table. Si vous disposez d'un contrôle d'accès basé sur Lake Formation sur une table, consultez la table pour vous assurer que les propriétés ne contiennent aucune donnée ou information sensible.
- Les clusters Amazon EMR dotés de Lake Formation ne prennent pas en charge le retour de Spark à HDFS lorsque Spark collecte les statistiques des tables. Cela permet généralement d'optimiser les performances des requêtes.

- Les opérations qui prennent en charge les contrôles d’accès basés sur Lake Formation avec les tables Apache Spark non gouvernées incluent `INSERT INTO` et `INSERT OVERWRITE`.
- Les opérations qui prennent en charge les contrôles d'accès basés sur Lake Formation avec Apache Spark et Apache Hive incluent `SELECT`, `DESCRIBE`, `SHOW DATABASE`, `SHOW TABLE`, `SHOW COLUMN` et `SHOW PARTITION`.
- Amazon EMR ne prend pas en charge le contrôle de l’accès aux opérations suivantes basées sur Lake Formation :
  - Écritures dans des tables gouvernées
  - Amazon EMR ne prend pas en charge `CREATE TABLE`. La version 6.10.0 et les versions ultérieures d’Amazon EMR prennent en charge `ALTER TABLE`.
  - Instructions DML autres que des commandes `INSERT`.
- Il existe des différences de performances entre la même requête avec et sans le contrôle d'accès basé sur Lake Formation.

Intégration d'Amazon EMR avec Apache Ranger

À partir d'Amazon EMR 5.32.0, vous pouvez lancer un cluster qui s'intègre nativement à Apache Ranger. Apache Ranger est un cadre open source permettant d'activer, de surveiller et de gérer la sécurité globale des données sur la plateforme Hadoop. Pour plus d'informations, consultez Apache Ranger. Grâce à l'intégration native, vous pouvez utiliser votre propre Apache Ranger pour appliquer un contrôle précis de l'accès aux données sur Amazon EMR.

Cette section fournit une présentation conceptuelle de l'intégration d'Amazon EMR à Apache Ranger. Elle comprend également les conditions préalables et les étapes nécessaires au lancement d'un cluster Amazon EMR intégré à Apache Ranger.

L'intégration native d'Amazon EMR avec Apache Ranger offre les principaux avantages suivants :

- Contrôle d'accès précis aux bases de données et aux tables Hive Metastore, qui vous permet de définir des politiques de filtrage des données au niveau de la base de données, de la table et de la colonne pour les applications Apache Spark et Apache Hive. Le filtrage au niveau des lignes et le masquage des données sont pris en charge par les applications Hive.
- La possibilité d'utiliser vos politiques Hive existantes directement avec Amazon EMR pour les applications Hive.

- Contrôle d'accès aux données Amazon S3 au niveau du préfixe et de l'objet, qui vous permet de définir des politiques de filtrage des données pour accéder aux données S3 à l'aide du système de fichiers EMR.
- Possibilité d'utiliser les CloudWatch journaux pour un audit centralisé.
- Amazon EMR installe et gère les plug-ins Apache Ranger en votre nom.

Apache Ranger

Apache Ranger est un cadre permettant d'activer, de surveiller et de gérer la sécurité globale des données sur la plateforme Hadoop.

Apache Ranger présente les caractéristiques suivantes :

- Administration de sécurité centralisée pour gérer toutes les tâches liées à la sécurité dans une interface utilisateur centrale ou à l'aide d'API REST.
- Autorisation précise pour effectuer une action ou une opération spécifique avec un composant ou un outil Hadoop, gérée via un outil d'administration central.
- Une méthode d'autorisation standardisée pour tous les composants Hadoop.
- Support amélioré pour les différentes méthodes d'autorisation.
- Audit centralisé de l'accès des utilisateurs et des actions administratives (liées à la sécurité) au sein de tous les composants de Hadoop.

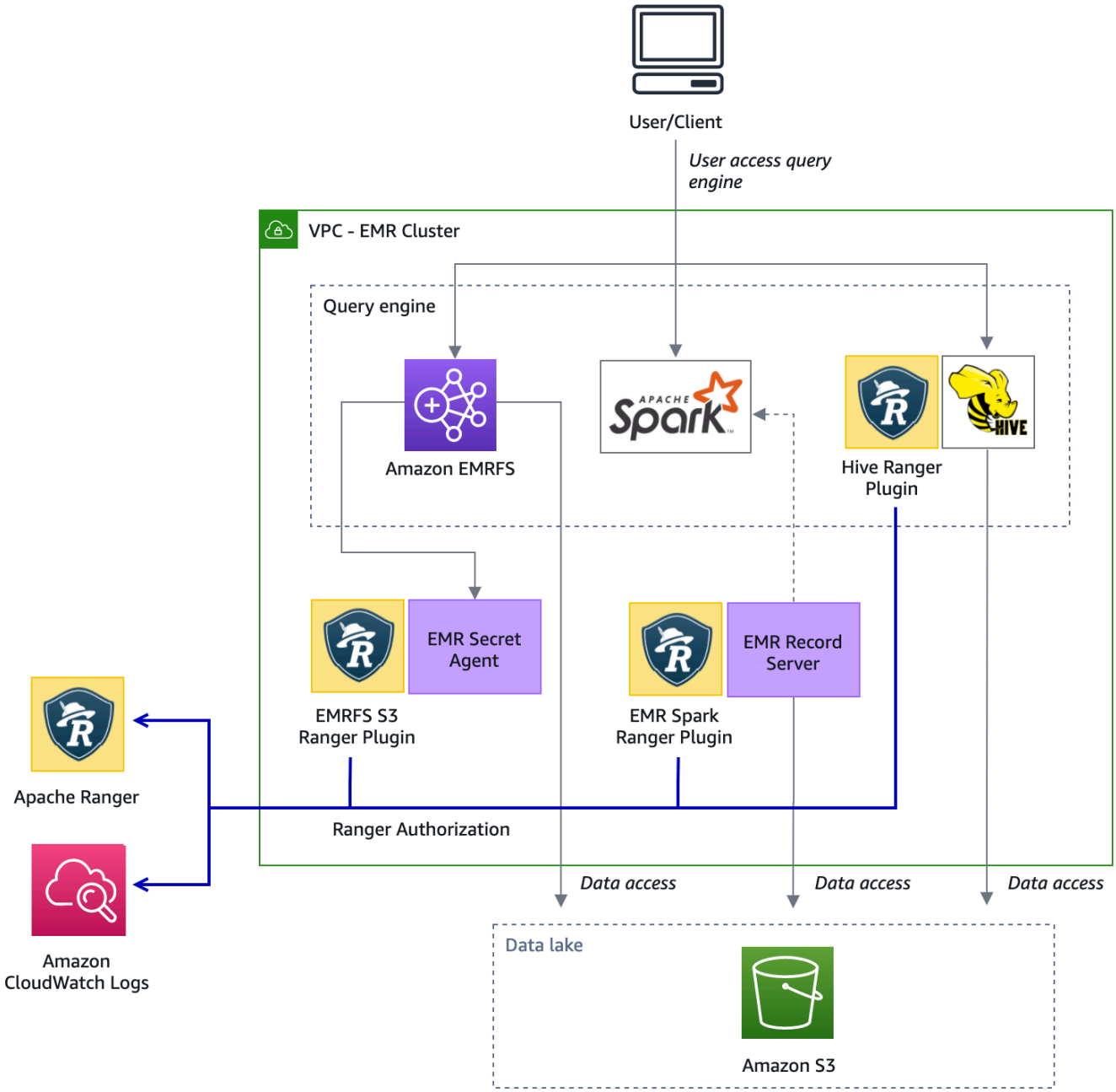
Apache Ranger utilise deux composants clés pour l'autorisation :

- Serveur d'administration des politiques Apache Ranger : ce serveur vous permet de définir les politiques d'autorisation pour les applications Hadoop. Lors de l'intégration à Amazon EMR, vous pouvez définir et appliquer des politiques permettant à Apache Spark et Hive d'accéder à Hive Metastore et au système de fichiers EMR (EMRFS) de données Amazon S3. Vous pouvez configurer un nouveau serveur d'administration des politiques Apache Ranger ou utiliser un serveur existant pour l'intégrer à Amazon EMR.
- Plug-in Apache Ranger : ce plug-in valide l'accès d'un utilisateur par rapport aux politiques d'autorisation définies sur le serveur d'administration des politiques Apache Ranger. Amazon EMR installe et configure le plug-in Apache Ranger automatiquement pour chaque application Hadoop sélectionnée dans la configuration d'Apache Ranger.

Rubriques

- Architecture de l'intégration d'Amazon EMR à Apache Ranger
- Composants d'Amazon EMR

Architecture de l'intégration d'Amazon EMR à Apache Ranger



Composants d'Amazon EMR

Amazon EMR permet un contrôle d'accès précis avec Apache Ranger par le biais des composants suivants. Consultez le schéma d'architecture pour une représentation visuelle de ces composants Amazon EMR avec les plug-ins Apache Ranger.

Agent secret – L'agent secret stocke les secrets en toute sécurité et les distribue à d'autres composants ou applications Amazon EMR. 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 nœud du cluster et intercepte les appels au service de métadonnées d'instance. Pour les demandes adressées aux informations d'identification du rôle du profil d'instance, l'agent secret transmet les informations d'identification en fonction de l'utilisateur demandeur et des ressources demandées après avoir autorisé la demande avec le plug-in EMRFS S3 Ranger. L'agent secret s'exécute en tant qu'utilisateur *`emrsecretagent`* et écrit des journaux dans le répertoire /emr/secretagent/log. Le processus s'appuie sur un ensemble spécifique de règles `iptables` pour fonctionner. Il est important de veiller à ce que `iptables` soit pas désactivé. Si vous personnalisez la configuration `iptables`, les règles de la table NAT doivent être préservées et laissées inchangées.

Serveur d'enregistrement EMR – Le serveur d'enregistrement reçoit des demandes d'accès aux données de la part de Spark. Il autorise ensuite les demandes en transférant les ressources demandées au plug-in Spark Ranger pour Amazon EMR. Le serveur d'enregistrement lit les données d'Amazon S3 et renvoie des données filtrées auxquelles l'utilisateur est autorisé à accéder conformément à la politique de Ranger. Le serveur d'enregistrement s'exécute sur chaque nœud du cluster en tant qu'utilisateur emr_record_server et écrit les journaux dans le répertoire /var/log/emr-record-server.

Prise en charge des applications et limitations

Applications prises en charge

L'intégration entre Amazon EMR et Apache Ranger dans laquelle EMR installe les plug-ins Ranger prend actuellement en charge les applications suivantes :

- Apache Spark (disponible avec EMR 5.32+ et EMR 6.3+)
- Apache Hive (disponible avec EMR 5.32+ et EMR 6.3+)
- S3 Access via EMRFS (disponible avec EMR 5.32+ et EMR 6.3+)

Les applications suivantes peuvent être installées sur un cluster EMR et devront peut-être être configurées pour répondre à vos besoins en matière de sécurité :

- Apache Hadoop (disponible avec EMR 5.32+ et EMR 6.3+, y compris YARN et HDFS)
- Apache Livy (disponible avec EMR 5.32+ et EMR 6.3+)
- Apache Zeppelin (disponible avec EMR 5.32+ et EMR 6.3+)
- Apache Hue (disponible avec EMR 5.32+ et EMR 6.3+)
- Ganglia (disponible avec EMR 5.32+ et EMR 6.3+)
- HCatalog (disponible avec EMR 5.32+ et EMR 6.3+)
- Mahout (disponible avec EMR 5.32+ et EMR 6.3+)
- MXNet (disponible avec EMR 5.32+ et EMR 6.3+)
- TensorFlow (Disponible avec EMR 5.32+ et EMR 6.3+)
- Tez (disponible avec EMR 5.32+ et EMR 6.3+)
- Trino (disponible avec EMR 6.7+)
- ZooKeeper (Disponible avec EMR 5.32+ et EMR 6.3+)

> **Important**
>
> Les applications répertoriées ci-dessus sont les seules actuellement prises en charge. Pour garantir la sécurité du cluster, vous êtes autorisé à créer un cluster EMR avec uniquement les applications de la liste ci-dessus lorsque Apache Ranger est activé.
> Les autres applications ne sont actuellement pas prises en charge. Pour garantir la sécurité de votre cluster, toute tentative d'installation d'autres applications entraînera le rejet de votre cluster.

Fonctionnalités prises en charge

Les fonctionnalités Amazon EMR suivantes peuvent être utilisées avec Amazon EMR et Apache Ranger :

- Chiffrement au repos et en transit
- Authentification Kerberos (obligatoire)
- Groupes d'instances, parcs d'instances et instances Spot

- Reconfiguration des applications sur un cluster en cours d'exécution
- chiffrement côté serveur (SSE) EMRFS

> **Note**
>
> Les paramètres de chiffrement d'Amazon EMR régissent le SSE. Pour plus d'informations, consultez Options de chiffrement.

Limites d'application

Il y a plusieurs limites à prendre en compte lorsque vous intégrez Amazon EMR et Apache Ranger :

- Vous ne pouvez actuellement pas utiliser la console pour créer une configuration de sécurité spécifiant l'option d'intégration de AWS Ranger dans le AWS GovCloud (US) Region. La configuration de la sécurité peut être effectuée à l'aide de la CLI.
- Kerberos doit être installé sur votre cluster.
- Les interfaces utilisateur des applications (interfaces utilisateur) telles que l'interface utilisateur YARN Resource Manager, l' NameNode interface utilisateur HDFS et l'interface utilisateur Livy ne sont pas définies avec l'authentification par défaut.
- Les autorisations par défaut du HDFS `umask` sont configurées de telle sorte que les objets créés sont définis sur `world wide readable` par défaut.
- Amazon EMR ne prend pas en charge le mode haute disponibilité (principal multiple) avec Apache Ranger.
- Pour connaître les limites supplémentaires, consultez les limites de chaque application.

> **Note**
>
> Les paramètres de chiffrement d'Amazon EMR régissent le SSE. Pour plus d'informations, consultez Options de chiffrement.

Limites de plug-in

Chaque plug-in possède des limites spécifiques. Pour connaître les limites du plug-in Apache Hive, consultez la section Limitations du plug-in Apache Hive. Pour connaître les limites du plug-in Apache

Spark, consultez la section Limitations du plug-in Apache Spark. Pour connaître les limites du plug-in EMRFS S3, consultez Limitations du plug-in EMRFS S3.

Configuration d'Amazon EMR pour Apache Ranger

Avant d'installer Apache Ranger, consultez les informations de cette section pour vous assurer qu'Amazon EMR est correctement configuré.

Rubriques

- Configuration du serveur d'administration Ranger
- Rôles IAM pour une intégration native avec Apache Ranger
- Création de la configuration de sécurité EMR
- Stockez les certificats TLS dans AWS Secrets Manager.
- Démarrez un cluster EMR.
- Configuration de Zeppelin pour les clusters Amazon EMR compatibles avec Apache Ranger
- Problèmes connus

Configuration du serveur d'administration Ranger

Pour l'intégration d'Amazon EMR, les plug-ins de l'application Apache Ranger doivent communiquer avec le serveur d'administration à l'aide du protocole TLS/SSL.

Prérequis : activation du protocole SSL pour le serveur d'administration Ranger

Apache Ranger sur Amazon EMR nécessite une communication SSL bidirectionnelle entre les plug-ins et le serveur d'administration Ranger. Pour garantir que les plugins communiquent avec le serveur Apache Ranger via SSL, activez l'attribut suivant dans le ranger-admin-site fichier .xml sur le serveur d'administration Ranger.

```
<property>
    <name>ranger.service.https.attrib.ssl.enabled</name>
    <value>true</value>
</property>
```

En outre, les configurations suivantes sont nécessaires.

```
<property>
    <name>ranger.https.attrib.keystore.file</name>
```

```
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.file</name>
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.pass</name>
    <value>_<KEYSTORE_PASSWORD>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.keyalias</name>
    <value><PRIVATE_CERTIFICATE_KEY_ALIAS></value>
</property>

<property>
    <name>ranger.service.https.attrib.clientAuth</name>
    <value>want</value>
</property>

<property>
    <name>ranger.service.https.port</name>
    <value>6182</value>
</property>
```

Certificat TLS

L'intégration d'Apache Ranger à Amazon EMR nécessite que le trafic entre les nœuds Amazon EMR et le serveur d'administration Ranger soit chiffré à l'aide du protocole TLS, et que les plug-ins Ranger s'authentifient auprès du serveur Apache Ranger à l'aide d'une authentification TLS mutuelle bidirectionnelle. Le service Amazon EMR a besoin du certificat public de votre serveur d'administration Ranger (spécifié dans l'exemple précédent) et du certificat privé.

Certificats du plug-in Apache Ranger

Les certificats TLS publics du plug-in Apache Ranger doivent être accessibles au serveur d'administration Apache Ranger pour valider la connexion des plug-ins. Il existe trois méthodes différentes pour ce faire.

Méthode 1 : configurer un magasin de confiance sur le serveur d'administration Apache Ranger

Renseignez les configurations suivantes dans le ranger-admin-site fichier .xml pour configurer un truststore.

```
<property>
    <name>ranger.truststore.file</name>
    <value><LOCATION TO TRUSTSTORE></value>
</property>

<property>
    <name>ranger.truststore.password</name>
    <value><PASSWORD FOR TRUSTSTORE></value>
</property>
```

Méthode 2 : charger le certificat dans le magasin de confiance cacerts de Java

Si votre serveur d'administration Ranger ne spécifie pas de magasin de confiance dans ses options JVM, vous pouvez placer les certificats publics du plug-in dans le magasin cacerts par défaut.

Méthode 3 : créer un magasin de confiance et le spécifier dans le cadre des options JVM

Dans `{RANGER_HOME_DIRECTORY}/ews/ranger-admin-services.sh`, modifiez `JAVA_OPTS` pour inclure "`-Djavax.net.ssl.trustStore=<TRUSTSTORE_LOCATION>`" et "`-Djavax.net.ssl.trustStorePassword=<TRUSTSTORE_PASSWORD>`". Par exemple, ajoutez la ligne suivante après le JAVA_OPTS existant.

```
JAVA_OPTS=" ${JAVA_OPTS} -Djavax.net.ssl.trustStore=${RANGER_HOME}/truststore/
truststore.jck -Djavax.net.ssl.trustStorePassword=changeit"
```

> **Note**
>
> Cette spécification peut exposer le mot de passe du magasin de confiance si un utilisateur parvient à se connecter au serveur d'administration Apache Ranger et à voir les processus en cours, par exemple lors de l'utilisation de la commande `ps`.

Utilisation des certificats auto-signés

Les certificats auto-signés ne sont pas recommandés en tant que certificats. Les certificats auto-signés ne peuvent pas être révoqués et peuvent ne pas être conformes aux exigences de sécurité internes.

Installation de définition de service

Une définition de service est utilisée par le serveur d'administration Ranger pour décrire les attributs des politiques d'une application. Les politiques sont ensuite stockées dans un référentiel de politiques que les clients peuvent télécharger.

Pour pouvoir configurer les définitions de service, des appels REST doivent être adressés au serveur d'administration Ranger. Consultez Apache Ranger PublicAPISv2 pour les API requises dans la section suivante.

Installation de la définition de service d'Apache Spark

Pour installer la définition de service d'Apache Spark, consultez Plug-in Apache Spark.

Installation de la définition du service EMRFS

Pour installer la définition du service S3 pour Amazon EMR, consultez Plug-in EMRFS S3.

Utilisation de la définition du service Hive

Apache Hive peut utiliser la définition de service Ranger existante fournie avec Apache Ranger 2.0 et versions ultérieures. Pour plus d'informations, consultez Plug-in Apache Hive.

Règles de trafic réseau

Lorsqu'Apache Ranger est intégré à votre cluster EMR, le cluster doit communiquer avec des serveurs supplémentaires et AWS.

Tous les nœuds Amazon EMR, y compris les nœuds principaux et les nœuds de tâches, doivent être en mesure de communiquer avec les serveurs d'administration Apache Ranger pour télécharger les politiques. Si votre administrateur Apache Ranger fonctionne sur Amazon EC2, vous devez mettre à jour le groupe de sécurité pour pouvoir prendre en charge le trafic du cluster EMR.

Outre la communication avec le serveur d'administration Ranger, tous les nœuds doivent être en mesure de communiquer avec les AWS services suivants :

- Amazon S3
- AWS KMS (si vous utilisez EMRFS SSE-KMS)
- Amazon CloudWatch
- AWS STS

Si vous envisagez d'exécuter votre cluster EMR dans un sous-réseau privé, configurez le VPC pour qu'il puisse communiquer avec ces services en utilisant soit AWS PrivateLink et les points de terminaison d'un VPC du Guide de l'utilisateur Amazon VPC, soit en utilisant l'instance de traduction d'adresses réseau (NAT) du Guide de l'utilisateur Amazon VPC.

Rôles IAM pour une intégration native avec Apache Ranger

L'intégration entre Amazon EMR et Apache Ranger repose sur trois rôles clés que vous devez créer avant de lancer votre cluster :

- Un profil d'instance Amazon EC2 personnalisé pour Amazon EMR
- Un rôle IAM pour les moteurs Apache Ranger
- Un rôle IAM pour les autres services AWS

Cette section présente ces rôles et les stratégies que vous devez inclure pour chaque rôle IAM. Pour plus d'informations sur la création de ces rôles, consultez Configuration du serveur d'administration Ranger.

Profil d'instance EC2

Amazon EMR utilise un rôle de service IAM pour effectuer des actions en votre nom afin d'allouer et de gérer les clusters. Le rôle de service pour les instances EC2 de cluster, également appelé profil d'instance EC2 pour Amazon EMR, est un type spécial de rôle de service attribué à chaque instance EC2 d'un cluster au moment du lancement.

Pour définir les autorisations d'interaction du cluster EMR avec les données Amazon S3 et avec le métastore Hive protégé par Apache Ranger et d'autres AWS services, définissez un profil d'instance EC2 personnalisé à utiliser à la place du `EMR_EC2_DefaultRole` profil lorsque vous lancez votre cluster.

Pour plus d'informations, consultez Rôle de service pour les instances EC2 de cluster (profil d'instance EC2) et Personnaliser les rôles IAM.

Vous devez ajouter les instructions suivantes au profil d'instance EC2 par défaut pour qu'Amazon EMR puisse baliser les sessions et accéder à celles qui stockent AWS Secrets Manager les certificats TLS.

```
{
  "Sid": "AllowAssumeOfRolesAndTagging",
  "Effect": "Allow",
```

```
      "Action": ["sts:TagSession", "sts:AssumeRole"],
      "Resource": [
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_ENGINE-
PLUGIN_DATA_ACCESS_ROLE_NAME>",
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_USER_ACCESS_ROLE_NAME>"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [

 "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<PLUGIN_TLS_SECRET_NAME>*",

 "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<ADMIN_RANGER_SERVER_TLS_SECRET_NAME>
        ]
    }
```

> **Note**
>
> Pour les autorisations du Secrets Manager, n'oubliez pas le caractère générique (« * ») à la fin du nom du secret, sinon vos demandes échoueront. Le caractère générique est destiné aux versions secrètes.

> **Note**
>
> Limitez le champ d'application de la AWS Secrets Manager politique aux seuls certificats requis pour le provisionnement.

Rôle IAM pour Apache Ranger

Ce rôle fournit des informations d'identification aux moteurs d'exécution fiables, tels qu'Apache Hive et Amazon EMR Record Server, pour accéder aux données Amazon S3. Utilisez uniquement ce rôle pour accéder aux données Amazon S3, y compris les clés KMS, si vous utilisez S3 SSE-KMS.

Ce rôle doit être créé avec la politique minimale indiquée dans l'exemple suivant.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudwatchLogsPermissions",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": [

 "arn:aws:logs:<REGION>:<AWS_ACCOUNT_ID>:<CLOUDWATCH_LOG_GROUP_NAME_IN_SECURITY_CONFIGURATION>:
      ]
    },
    {
      "Sid": "BucketPermissionsInS3Buckets",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1",
        "arn:aws:s3:::bucket2"*
        ]
    },
    {
      "Sid": "ObjectPermissionsInS3Objects",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1/*",
          "arn:aws:s3:::bucket2/*"
*        ]
    }
  ]
```

```
}
```

> **Important**
>
> L'astérisque « * » à la fin de la ressource de CloudWatch journal doit être inclus pour autoriser l'écriture dans les flux de journaux.

> **Note**
>
> Si vous utilisez la vue de cohérence EMRFS ou le chiffrement S3-SSE, ajoutez des autorisations aux tables DynamoDB et aux clés KMS afin que les moteurs d'exécution puissent interagir avec ces moteurs.

Le rôle IAM pour Apache Ranger est assumé par le rôle de profil d'instance EC2. Utilisez l'exemple suivant pour créer une politique d'approbation qui permet au rôle IAM pour Apache Ranger d'être assumé par le rôle de profil d'instance EC2.

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

Rôle IAM pour les autres services AWS

Ce rôle fournit aux utilisateurs qui ne sont pas des moteurs d'exécution fiables des informations d'identification leur permettant d'interagir avec AWS les services, si nécessaire. N'utilisez pas ce rôle IAM pour autoriser l'accès aux données Amazon S3, sauf s'il s'agit de données qui devraient être accessibles à tous les utilisateurs.

Ce rôle sera assumé par le rôle de profil d'instance EC2. Utilisez l'exemple suivant pour créer une politique d'approbation qui permet au rôle IAM pour Apache Ranger d'être assumé par le rôle de profil d'instance EC2.

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

Validation de vos autorisations

Consultez Résolution des problèmes liés à Apache pour des instructions relatives à la validation des autorisations.

Création de la configuration de sécurité EMR

Création d'une configuration de sécurité Amazon EMR pour Apache Ranger

Avant de lancer un cluster Amazon EMR intégré à Apache Ranger, créez une configuration de sécurité.

Console

Pour créer une configuration de sécurité qui spécifie l'option Intégration AWS Ranger

1. Dans la console Amazon EMR, sélectionnez Configurations de sécurité, puis Créer.
2. 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.
3. Sous Intégration AWS Ranger, sélectionnez Activer le contrôle précis des accès géré par Apache Ranger.
4. Sélectionnez votre Rôle IAM pour Apache Ranger à appliquer. Pour plus d'informations, consultez Rôles IAM pour une intégration native avec Apache Ranger.
5. Sélectionnez votre rôle IAM pour les autres services AWS à appliquer.
6. Configurez les plug-ins pour vous connecter au serveur d'administration Ranger en saisissant l'ARN de Secret Manager pour le serveur d'administration et l'adresse.
7. Sélectionnez les applications pour configurer les plug-ins Ranger. Renseignez l'ARN du Secret Manager qui contient le certificat TLS privé du plug-in.

Si vous ne configurez pas Apache Spark ou Apache Hive et qu'ils sont sélectionnés comme application pour votre cluster, la demande échoue.

8. Définissez les autres options de la configuration de sécurité selon vos besoins et choisissez Create (Créer). Vous devez activer l'authentification Kerberos à l'aide du KDC dédié au cluster ou externe.

> **Note**
>
> Vous ne pouvez actuellement pas utiliser la console pour créer une configuration de sécurité spécifiant l'option d'intégration de AWS Ranger dans le AWS GovCloud (US) Region. La configuration de la sécurité peut être effectuée à l'aide de la CLI.

CLI

Pour créer une configuration de sécurité pour l'intégration d'Apache Ranger

1. Remplacez *<ACCOUNT ID>* par votre identifiant de AWS compte.
2. Remplacez *<REGION>* par la région dans laquelle se trouve la ressource.
3. Spécifiez une valeur pour `TicketLifetimeInHours` afin de déterminer la période pendant laquelle un ticket Kerberos émis par le KDC est valide.
4. Spécifiez l'adresse du serveur d'administration Ranger pour `AdminServerURL`.

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24
            }
        }
    },
    "AuthorizationConfiguration":{
      "RangerConfiguration":{
         "AdminServerURL":"https://_<RANGER ADMIN SERVER IP>_:6182",
         "RoleForRangerPluginsARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<RANGER PLUGIN
 DATA ACCESS ROLE NAME>_",
```

```
        "RoleForOtherAWSServicesARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<USER
ACCESS ROLE NAME>_",
        "AdminServerSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES ADMIN SERVERS PUBLIC TLS CERTIFICATE
WITHOUT VERSION>_",
        "RangerPluginConfigurations":[
           {
              "App":"Spark",
              "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES SPARK PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
              "PolicyRepositoryName":"<SPARK SERVICE NAME eg. amazon-emr-spark>"
           },
           {
              "App":"Hive",
              "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES Hive PLUGIN PRIVATE TLS CERTIFICATE WITHOUT
VERSION>_",
              "PolicyRepositoryName":"<HIVE SERVICE NAME eg. Hivedev>"
           },
           {
              "App":"EMRFS-S3",
              "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES EMRFS S3 PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
              "PolicyRepositoryName":"<EMRFS S3 SERVICE NAME eg amazon-emr-emrfs>"
           },
      {
              "App":"Trino",
              "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES TRINO PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
              "PolicyRepositoryName":"<TRINO SERVICE NAME eg amazon-emr-trino>"
           }
        ],
        "AuditConfiguration":{
           "Destinations":{
              "AmazonCloudWatchLogs":{
                 "CloudWatchLogGroup":"arn:aws:logs:<REGION>:_<ACCOUNT ID>_:log-
group:_<LOG GROUP NAME FOR AUDIT EVENTS>_"
              }
           }
        }
     }
```

```
    }
}
```

Ce PolicyRespositoryNames sont les noms de service spécifiés dans votre interface d'administration Apache Ranger.

Créez une configuration de sécurité Amazon EMR à l'aide de la commande suivante. Remplacez la configuration de sécurité par le nom de votre choix. Sélectionnez cette configuration par son nom lorsque vous créez votre cluster.

```
aws emr create-security-configuration \
--security-configuration file://./security-configuration.json \
--name security-configuration
```

Configuration de fonctionnalités de sécurité supplémentaires

Pour intégrer en toute sécurité Amazon EMR à Apache Ranger, configurez les fonctions de sécurité EMR suivantes :

- Activez l'authentification Kerberos à l'aide du KDC dédié au cluster ou externe. Pour obtenir des instructions, veuillez consulter Utilisation de Kerberos pour l'authentification avec Amazon EMR.
- (Facultatif) Activez le chiffrement en transit ou au repos. Pour plus d'informations, consultez Options de chiffrement.

Pour plus d'informations, consultez Sécurité dans Amazon EMR.

Stockez les certificats TLS dans AWS Secrets Manager.

Les plug-ins Ranger installés sur un cluster Amazon EMR et le serveur d'administration Ranger doivent communiquer via TLS pour garantir que les données de politique et autres informations envoyées ne peuvent pas être lues en cas d'interception. EMR exige également que les plug-ins s'authentifient auprès du serveur d'administration Ranger en fournissant son propre certificat TLS et en effectuant une authentification TLS bidirectionnelle. Cette configuration a nécessité la création de quatre certificats : deux paires de certificats TLS privés et publics. Pour obtenir des instructions sur l'installation du certificat sur votre serveur d'administration Ranger, consultez Configuration du serveur d'administration Ranger. Pour terminer la configuration, les plug-ins Ranger installés sur le cluster EMR ont besoin de deux certificats : le certificat TLS public de votre serveur d'administration et le certificat privé que le plug-in utilisera pour s'authentifier auprès du serveur d'administration

Ranger. Pour fournir ces certificats TLS, ils doivent figurer dans AWS Secrets Manager et fournis dans une configuration de sécurité EMR.

> **Note**
>
> Il est fortement recommandé, mais pas obligatoire, de créer une paire de certificats pour chacune de vos applications afin de limiter l'impact si l'un des certificats du plug-in est compromis.

> **Note**
>
> Vous devez suivre et alterner les certificats avant leur date d'expiration.

Format du certificat

L'importation des certificats vers le AWS Secrets Manager est la même, qu'il s'agisse du certificat du plugin privé ou du certificat d'administrateur Ranger public. Avant d'importer les certificats TLS, ceux-ci doivent être au format PEM 509x.

Voici un exemple de certificat public au format :

```
-----BEGIN CERTIFICATE-----
...Certificate Body...
-----END CERTIFICATE-----
```

Voici un exemple de certificat privé au format :

```
-----BEGIN PRIVATE KEY-----
...Private Certificate Body...
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
...Trust Certificate Body...
-----END CERTIFICATE-----
```

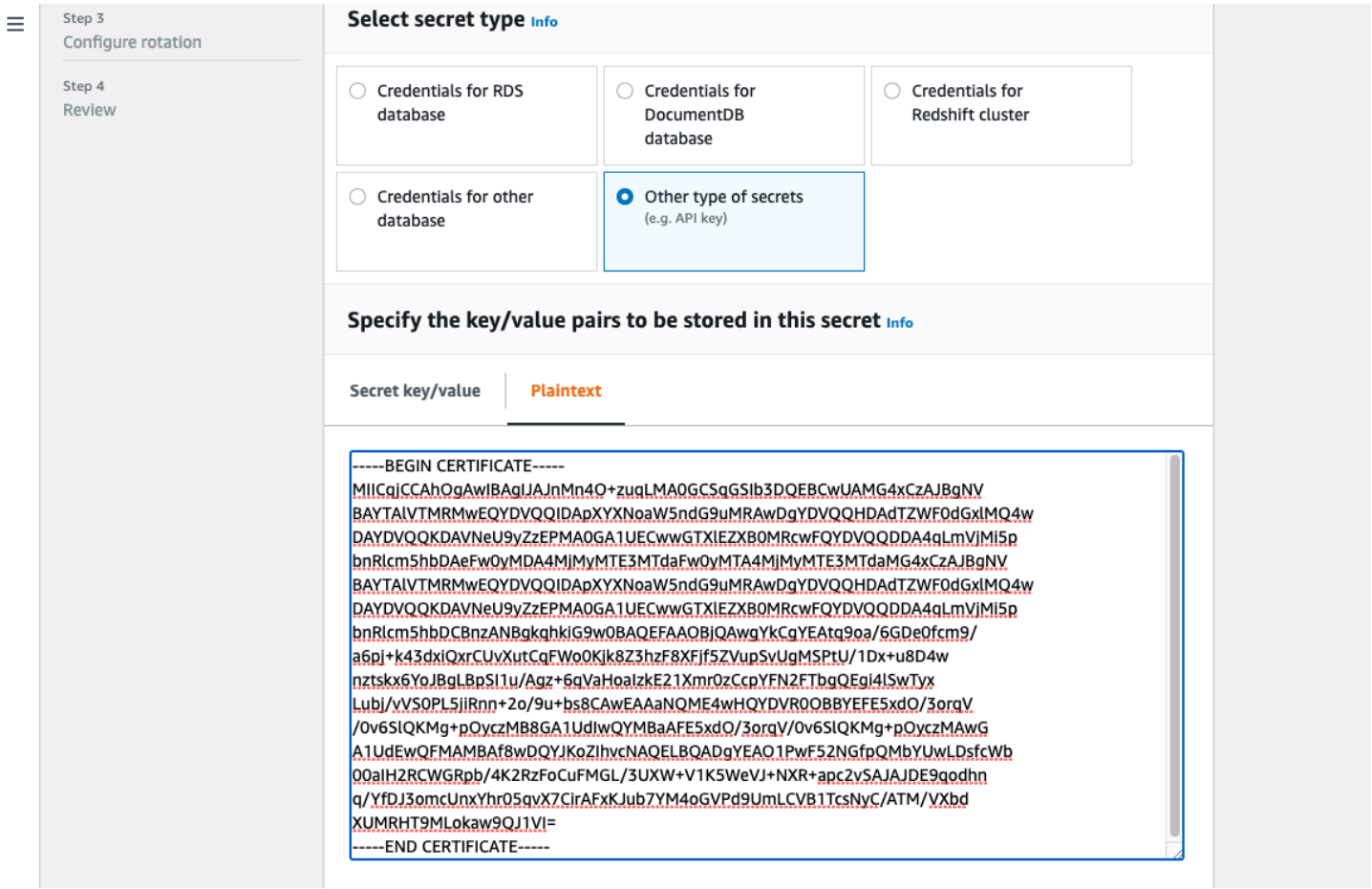
Le certificat privé doit également contenir un certificat de confiance.

Vous pouvez vérifier que le format des certificats est correct en exécutant la commande suivante :

```
openssl x509 -in <PEM FILE> -text
```

Importation d'un certificat dans AWS Secrets Manager

Lorsque vous créez votre secret dans Secrets Manager, choisissez Autre type de secrets sous Type de secret et collez votre certificat codé PEM dans le champ Texte en clair.



Démarrez un cluster EMR.

Avant de lancer un cluster Amazon EMR avec Apache Ranger, assurez-vous que chaque composant répond aux exigences de version minimale suivantes :

- Amazon EMR 5.32.0 ou version ultérieure, ou 6.3.0 ou version ultérieure. Nous vous recommandons de choisir la dernière version Amazon EMR.
- Serveur d'administration Apache Ranger 2.x.

Procédez comme suit.

- Installez Apache Ranger si ce n'est pas déjà fait. Pour plus d'informations, consultez Installation d'Apache Ranger 0.5.0.
- Assurez-vous qu'il existe une connectivité réseau entre votre cluster Amazon EMR et le serveur d'administration Apache Ranger. Consultez Configuration du serveur d'administration Ranger.
- Créez les rôles IAM nécessaires. veuillez consulter Rôles IAM pour une intégration native avec Apache Ranger.
- Créez une configuration de sécurité EMR pour l'installation d'Apache Ranger. Pour plus d'informations, consultez Création de la configuration de sécurité EMR.

Configuration de Zeppelin pour les clusters Amazon EMR compatibles avec Apache Ranger

Cette rubrique explique comment configurer Apache Zeppelin pour un cluster Amazon EMR compatible avec Apache Ranger afin que vous puissiez utiliser Zeppelin comme bloc-notes pour une exploration interactive des données. Zeppelin est inclus dans les version 5.0.0 et ultérieures d'Amazon EMR. Les versions antérieures incluent Zeppelin en tant qu'application d'environnement de test (sandbox). Pour plus d'informations, consultez Versions 4.x d'Amazon EMR dans le Guide de version Amazon EMR.

Par défaut, Zeppelin est configuré avec un identifiant et un mot de passe par défaut qui ne sont pas sécurisés dans un environnement multi-locataire.

Pour configurer Zeppelin, effectuez les opérations suivantes.

1. Modifiez le mécanisme d'authentification.

   Modifiez le fichier `shiro.ini` pour implémenter votre mécanisme d'authentification préféré. Zeppelin prend en charge Active Directory, LDAP, PAM et Knox SSO. Consultez Authentification Apache Shiro pour Apache Zeppelin pour plus d'informations.
2. Configuration de Zeppelin pour qu'il se fasse passer pour l'utilisateur final

   Lorsque vous autorisez Zeppelin à se faire passer pour l'utilisateur final, les tâches soumises par Zeppelin peuvent être exécutées en tant qu'utilisateur final. Ajoutez la configuration suivante à `core-site.xml` :

```
[
  {
```

```
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.zeppelin.hosts": "*",
      "hadoop.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Ensuite, ajoutez la configuration suivante à `hadoop-kms-site.xml` dans `/etc/hadoop/conf` :

```
[
  {
    "Classification": "hadoop-kms-site",
    "Properties": {
      "hadoop.kms.proxyuser.zeppelin.hosts": "*",
      "hadoop.kms.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Vous pouvez également ajouter ces configurations à votre cluster Amazon EMR à l'aide de la console en suivant les étapes décrites dans Reconfigurer un groupe d'instances dans la console.

3. Autoriser Zeppelin à sudo en tant qu'utilisateur final

   Créez un fichier `/etc/sudoers.d/90-zeppelin-user` contenant ce qui suit :

   ```
   zeppelin ALL=(ALL) NOPASSWD:ALL
   ```

4. Modifiez les paramètres des interpréteurs pour exécuter les tâches des utilisateurs dans leurs propres processus.

   Pour tous les interpréteurs, configurez-les pour instancier les interpréteurs « par utilisateur » dans des processus « isolés ».

5. Modifier **`zeppelin-env.sh`**

   Ajoutez ce qui suit à `zeppelin-env.sh` pour que Zeppelin lance les interprètes en tant qu'utilisateur final :

   ```
   ZEPPELIN_IMPERSONATE_USER=`echo ${ZEPPELIN_IMPERSONATE_USER} | cut -d @ -f1`
   export ZEPPELIN_IMPERSONATE_CMD='sudo -H -u ${ZEPPELIN_IMPERSONATE_USER} bash -c'
   ```

   Ajoutez ce qui suit à `zeppelin-env.sh` pour modifier les autorisations par défaut du bloc-notes en lecture seule pour le créateur uniquement :

   ```
   export ZEPPELIN_NOTEBOOK_PUBLIC="false"
   ```

   Enfin, ajoutez ce qui suit `zeppelin-env.sh` pour inclure le chemin de RecordServer classe EMR après la première `CLASSPATH` instruction :

   ```
   export CLASSPATH="$CLASSPATH:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-connector-common.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-spark-connector.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-client.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-common.jar:/usr/share/aws/emr/record-server/lib/jars/secret-agent-interface.jar"
   ```

6. Redémarrez Zeppelin.

   Exécutez la commande suivante pour redémarrer Zeppelin :

   ```
   sudo systemctl restart zeppelin
   ```

Problèmes connus

Problèmes connus

Il existe un problème connu dans la version 5.32 d'Amazon EMR, dans lequel les autorisations pour `hive-site.xml` ont été modifiées afin que seuls les utilisateurs privilégiés puissent le lire, car des informations d'identification peuvent y être stockées. Cela pourrait empêcher Hue de lire `hive-site.xml` et provoquer le rechargement continu des pages Web. Si vous rencontrez ce problème, ajoutez la configuration suivante pour le résoudre :

```
[
  {
    "Classification": "hue-ini",
    "Properties": {},
    "Configurations": [
      {
        "Classification": "desktop",
        "Properties": {
          "server_group":"hive_site_reader"
         },
        "Configurations":[
        ]
      }
    ]
  }
]
```

Il existe un problème connu selon lequel le plug-in EMRFS S3 pour Apache Ranger ne prend actuellement pas en charge la fonctionnalité de zone de sécurité d'Apache Ranger. Les restrictions de contrôle d'accès définies à l'aide de la fonctionnalité Zone de sécurité ne sont pas appliquées sur vos clusters Amazon EMR.

Interfaces utilisateur des applications

Par défaut, l'interface utilisateur de l'application n'effectue pas d'authentification. Cela inclut l'ResourceManager interface utilisateur, l' NodeManager interface utilisateur, l'interface utilisateur Livy, entre autres. En outre, tout utilisateur ayant la possibilité d'accéder aux interfaces utilisateur peut consulter les informations relatives aux tâches de tous les autres utilisateurs.

Si ce comportement n'est pas souhaité, vous devez vous assurer qu'un groupe de sécurité est utilisé pour restreindre l'accès des utilisateurs aux interfaces utilisateur des applications.

Autorisations par défaut de HDFS

Par défaut, les objets créés par les utilisateurs dans HDFS reçoivent des autorisations lisibles par tout le monde. Cela peut potentiellement rendre les données lisibles par des utilisateurs qui ne devraient

pas y avoir accès. Pour modifier ce comportement de telle sorte que les autorisations de fichier par défaut soient définies pour lire et écrire uniquement par le créateur de la tâche, effectuez ces étapes.

Lors de la création de votre cluster EMR, fournissez la configuration suivante :

```
[
  {
    "Classification": "hdfs-site",
    "Properties": {
      "dfs.namenode.acls.enabled": "true",
      "fs.permissions.umask-mode": "077",
      "dfs.permissions.superusergroup": "hdfsadmingroup"
    }
  }
]
```

Exécutez également l'action d'amorçage suivante :

```
--bootstrap-actions Name='HDFS UMask Setup',Path=s3://elasticmapreduce/hdfs/umask/
umask-main.sh
```

Plug-ins Apache Ranger

Les plug-ins Apache Ranger valident l'accès d'un utilisateur par rapport aux politiques d'autorisation définies sur le serveur d'administration des politiques Apache Ranger.

Rubriques

- Plug-in Apache Hive
- Plug-in Apache Spark
- Plug-in EMRFS S3
- Plug-in Trino

Plug-in Apache Hive

Apache Hive est un moteur d'exécution populaire au sein de l'écosystème Hadoop. Amazon EMR fournit un plug-in Apache Ranger permettant de fournir des contrôles d'accès précis pour Hive. Le plug-in est compatible avec le serveur d'administration open source Apache Ranger version 2.0 et ultérieure.

Rubriques

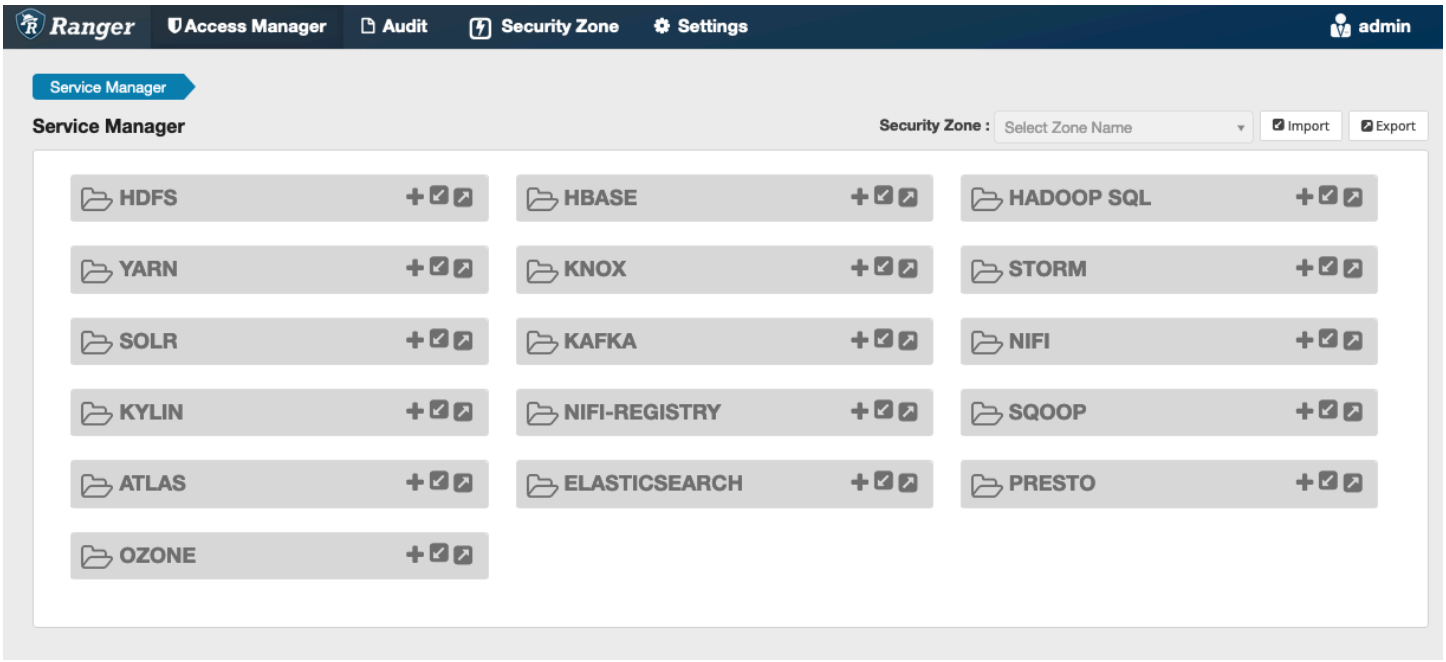
- Fonctionnalités prises en charge
- Installation de la configuration du service
- Considérations
- Limites

Fonctionnalités prises en charge

Le plug-in Apache Ranger pour Hive on EMR prend en charge toutes les fonctionnalités du plug-in open source, notamment les contrôles d'accès aux bases de données, aux tables et aux colonnes, le filtrage des lignes et le masquage des données. Pour un tableau des commandes Hive et des autorisations Ranger associées, consultez Mappage descommandes Hive aux autorisations Ranger.

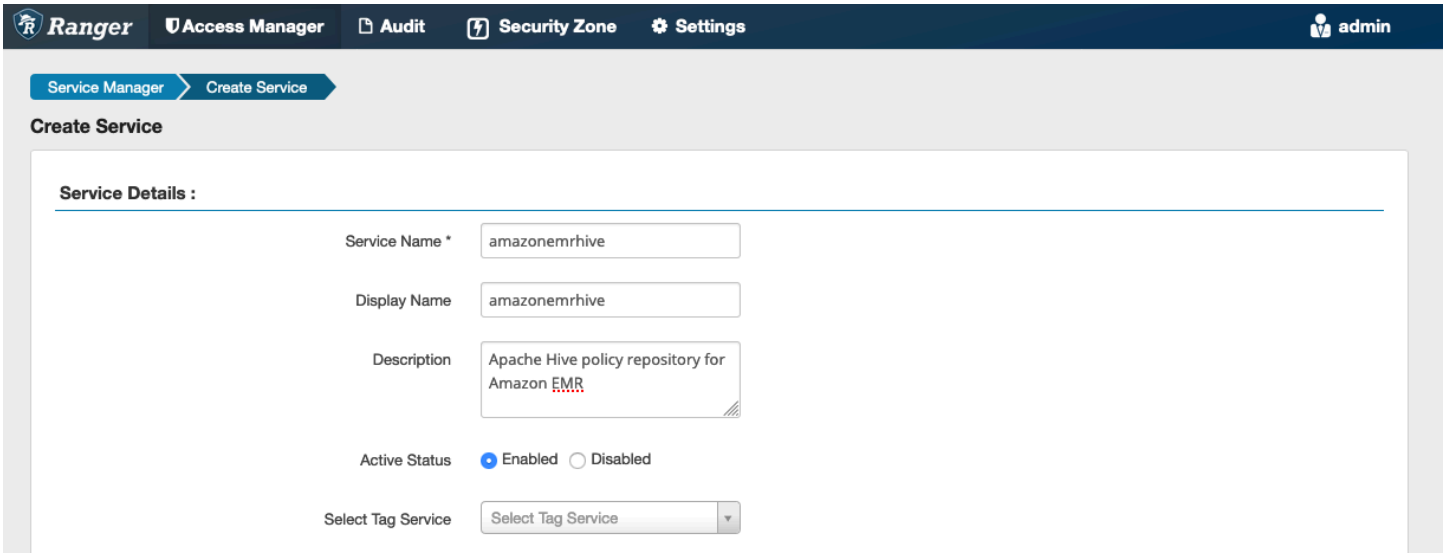
Installation de la configuration du service

Le plug-in Apache Hive est compatible avec la définition de service Hive existante dans Apache Hive Hadoop SQL.



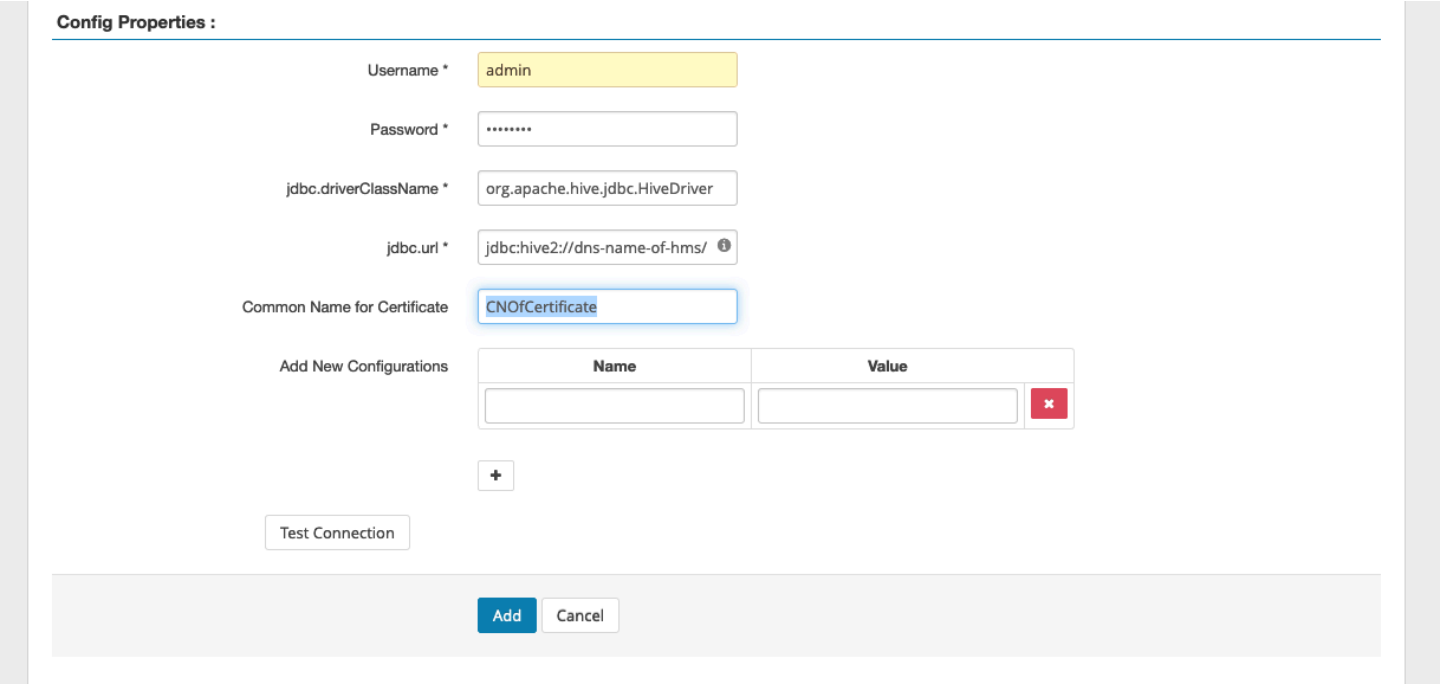
Si vous ne possédez pas d'instance du service sous Hadoop SQL, comme indiqué ci-dessus, vous pouvez en créer une. Cliquez sur le signe + à côté de Hadoop SQL.

1. Nom du service (s'il est affiché) : entrez le nom du service. La valeur suggérée est **`amazonemrhive`**. Notez le nom de ce service : il est nécessaire lors de la création d'une configuration de sécurité EMR.
2. Nom d'affichage : entrez le nom à afficher pour le service. La valeur suggérée est **`amazonemrhive`**.

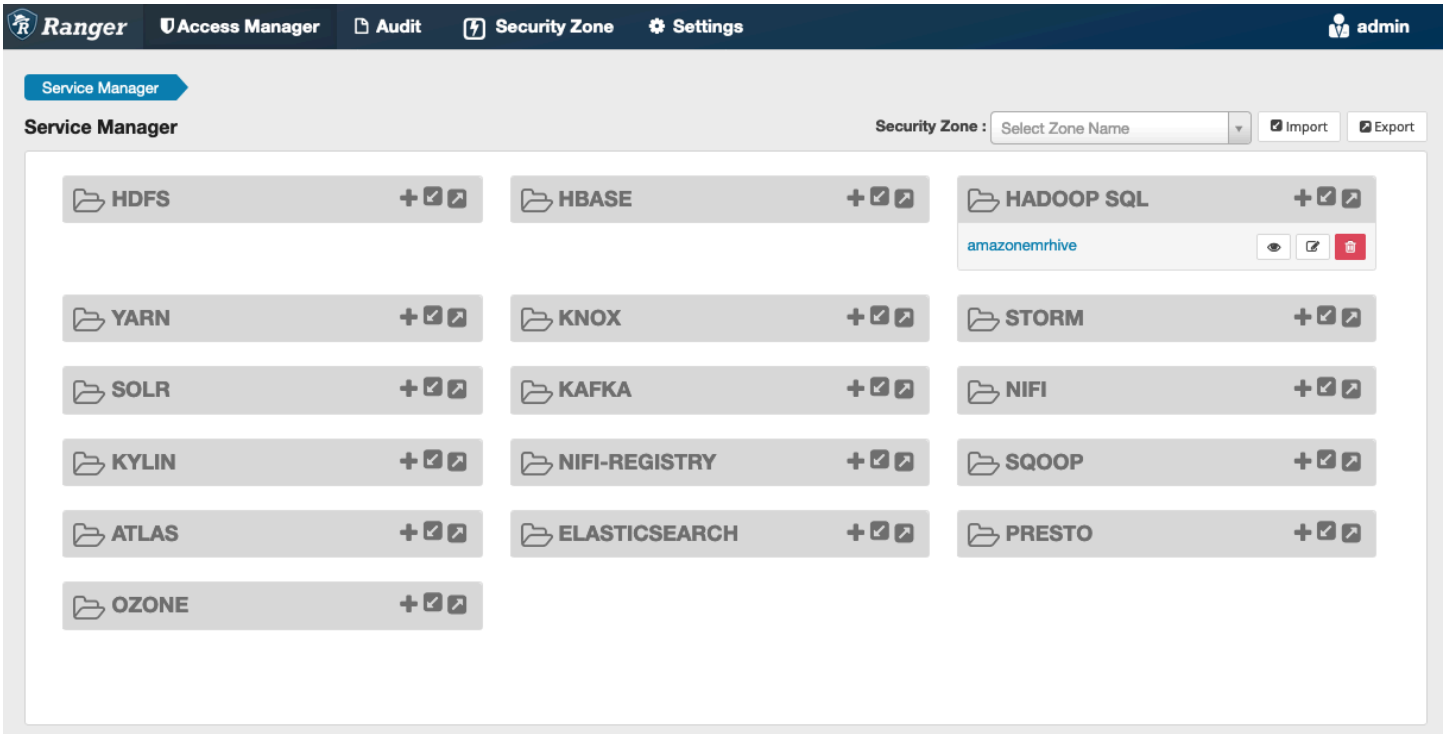


Les propriétés de configuration d'Apache Hive sont utilisées pour établir une connexion avec votre serveur d'administration Apache Ranger avec un HiveServer 2 pour implémenter la saisie automatique lors de la création de politiques. Il n'est pas nécessaire que les propriétés ci-dessous soient exactes si vous ne disposez pas d'un processus HiveServer 2 persistant et peuvent être renseignées avec n'importe quelle information.

- Nom d'utilisateur : entrez un nom d'utilisateur pour la connexion JDBC à une instance d'une instance HiveServer 2.
- Mot de passe : entrez le mot de passe pour le nom d'utilisateur ci-dessus.
- jdbc.driver. ClassName: Entrez le nom de classe de la classe JDBC pour la connectivité Apache Hive. La valeur par défaut peut être utilisée.
- jdbc.url : Entrez la chaîne de connexion JDBC à utiliser lors de la connexion à 2. HiveServer
- Nom commun du certificat : champ CN du certificat utilisé pour se connecter au serveur d'administration à partir d'un plug-in client. Cette valeur doit correspondre au champ CN de votre certificat TLS créé pour le plug-in.



Le bouton Tester la connexion vérifie si les valeurs ci-dessus peuvent être utilisées pour établir une connexion réussie à l'instance HiveServer 2. Une fois le service créé avec succès, le gestionnaire de services devrait ressembler à ce qui suit :



Considérations

Serveur de métadonnées Hive

Le serveur de métadonnées Hive n'est accessible que par des moteurs fiables, en particulier Hive et `emr_record_server`, pour se protéger contre tout accès non autorisé. Le serveur de métadonnées Hive est également accessible par tous les nœuds du cluster. Le port 9083 requis permet à tous les nœuds d'accéder au nœud principal.

Authentification

Par défaut, Apache Hive est configuré pour s'authentifier à l'aide de Kerberos conformément à la configuration de sécurité EMR. HiveServer2 peuvent également être configurés pour authentifier les utilisateurs à l'aide de LDAP. Consultez Implémentation de l'authentification LDAP pour Hive sur un cluster Amazon EMR à locataires multiples pour plus d'informations.

Limites

Les limitations actuelles du plug-in Apache Hive sur Amazon EMR 5.x sont les suivantes :

- Les rôles Hive ne sont actuellement pas pris en charge. Les instructions Grant, Revoke ne sont pas prises en charge.
- La CLI Hive n'est pas prise en charge. JDBC/Beeline est le seul moyen autorisé de connecter Hive.
- La configuration `hive.server2.builtin.udf.blacklist` doit être remplie avec des UDF que vous jugez dangereux.

Plug-in Apache Spark

Amazon EMR a intégré l'EMR afin de fournir un contrôle d'accès précis RecordServer pour SparkSQL. L'EMR RecordServer est un processus privilégié qui s'exécute sur tous les nœuds d'un cluster compatible avec Apache Ranger. Lorsqu'un pilote ou un exécuteur Spark exécute une instruction SparkSQL, toutes les métadonnées et demandes de données passent par le. RecordServer Pour en savoir plus sur l'EMR RecordServer, consultez la Composants d'Amazon EMR page.

Rubriques

- Fonctionnalités prises en charge
- Redéployer la définition du service pour utiliser les instructions INSERT, ALTER ou DDL

- Installation de la définition du service
- Création de politiques SparkSQL
- Considérations
- Limites

Fonctionnalités prises en charge

Instruction SQL/action Ranger	STATUS	Version EMR pris en charge
SELECT	Pris en charge	À partir de 5.32
SHOW DATABASES	Pris en charge	À partir de 5.32
SHOW COLUMNS	Pris en charge	À partir de 5.32
SHOW TABLES	Pris en charge	À partir de 5.32
SHOW TABLE PROPERTIES	Pris en charge	À partir de 5.32
DESCRIBE TABLE	Pris en charge	À partir de 5.32
INSERT OVERWRITE	Pris en charge	À partir de 5.34 et 6.4
INSERT INTO	Pris en charge	À partir de 5.34 et 6.4
ALTER TABLE	Pris en charge	À partir de 6.4
CREATE TABLE	Pris en charge	Àpartir de 5.35 et 6.7
CREATE DATABASE	Pris en charge	Àpartir de 5.35 et 6.7
DROP TABLE	Pris en charge	Àpartir de 5.35 et 6.7

Instruction SQL/action Ranger	STATUS	Version EMR pris en charge
DROP DATABASE	Pris en charge	Àpartir de 5.35 et 6.7
DROP VIEW	Pris en charge	Àpartir de 5.35 et 6.7
CREATE VIEW	Non pris en charge	

Les fonctionnalités suivantes sont prises en charge lors de l'utilisation de SparkSQL :

- Un contrôle d'accès précis sur les tables du métastore Hive et des politiques peuvent être créées au niveau de la base de données, de la table et de la colonne.
- Les politiques Apache Ranger peuvent inclure des politiques d'octroi et de refus pour les utilisateurs et les groupes.
- Les événements d'audit sont soumis à CloudWatch Logs.

Redéployer la définition du service pour utiliser les instructions INSERT, ALTER ou DDL

> **Note**
>
> À partir d'Amazon EMR 6.4, vous pouvez utiliser Spark SQL avec les instructions : INSERT INTO, INSERT OVERWRITE ou ALTER TABLE. À partir d'Amazon EMR 6.7, vous pouvez utiliser Spark SQL pour créer ou supprimer des bases de données et des tables. Si vous disposez d'une installation existante sur le serveur Apache Ranger avec des définitions de service Apache Spark déployées, utilisez le code suivant pour redéployer les définitions de service.

```
# Get existing Spark service definition id calling Ranger REST API and JSON
 processor
curl --silent -f -u <admin_user_login>:<password_for_ranger_admin_user> \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef/
name/amazon-emr-spark' | jq .id

# Download the latest Service definition
```

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-spark.json

# Update the service definition using the Ranger REST API
curl -u <admin_user_login>:<password_for_ranger_admin_user> -X PUT -d @ranger-
servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/
servicedef/<Spark service definition id from step 1>'
```

Installation de la définition du service

L'installation de la définition du service Apache Spark d'EMR nécessite la configuration du serveur d'administration Ranger. veuillez consulter Configuration du serveur d'administration Ranger.

Pour installer la définition du service Apache Spark, procédez comme suit :

Étape 1 : connexion SSH au serveur d'administration Apache Ranger

Par exemple :

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Étape 2 : téléchargez la définition du service et le plug-in du serveur d'administration Apache Ranger

Dans un répertoire temporaire, téléchargez la définition de service. Cette définition de service est prise en charge par les versions 2.x de Ranger.

```
mkdir /tmp/emr-spark-plugin/
cd /tmp/emr-spark-plugin/

wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-spark-plugin-2.x.jar
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-spark.json
```

Étape 3 : installez le plug-in Apache Spark pour Amazon EMR

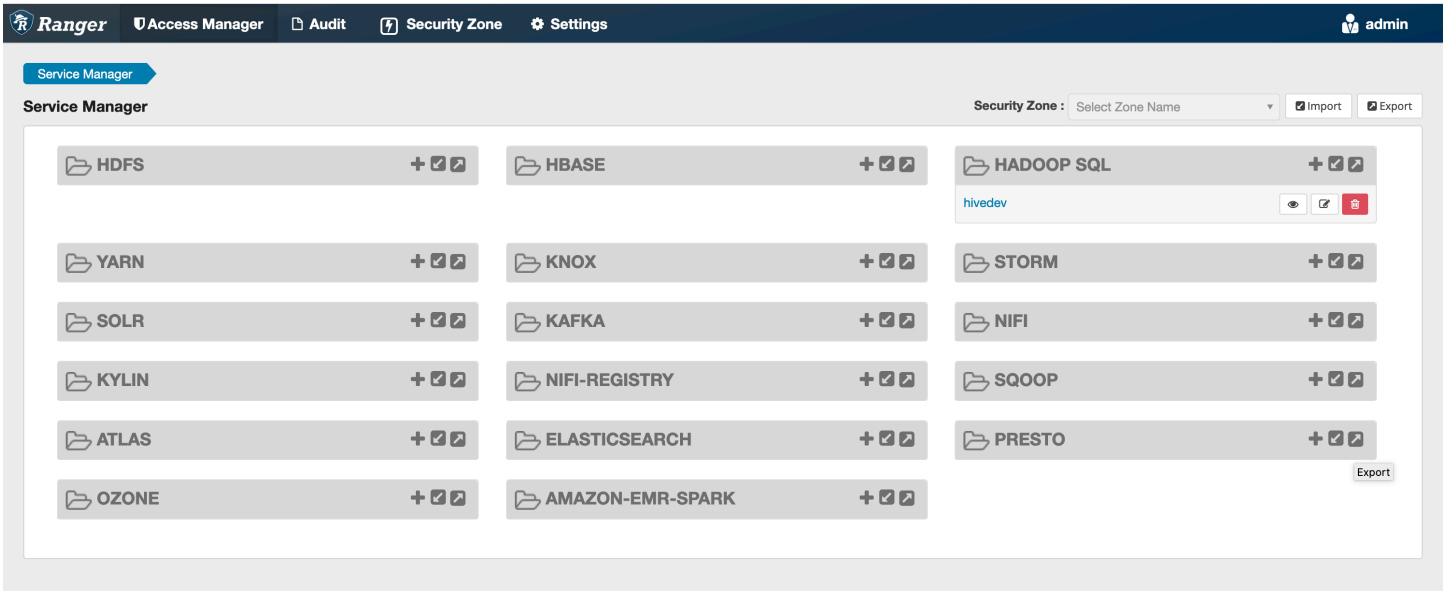
```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
```

```
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-spark
mv ranger-spark-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/
amazon-emr-spark
```

Étape 4 : enregistrez la définition du service Apache Spark pour Amazon EMR

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Si cette commande s'exécute correctement, un nouveau service appelé « AMAZON-EMR-SPARK » s'affiche dans votre interface utilisateur d'administration de Ranger, comme indiqué dans l'image suivante (la version 2.0 de Ranger est illustrée).

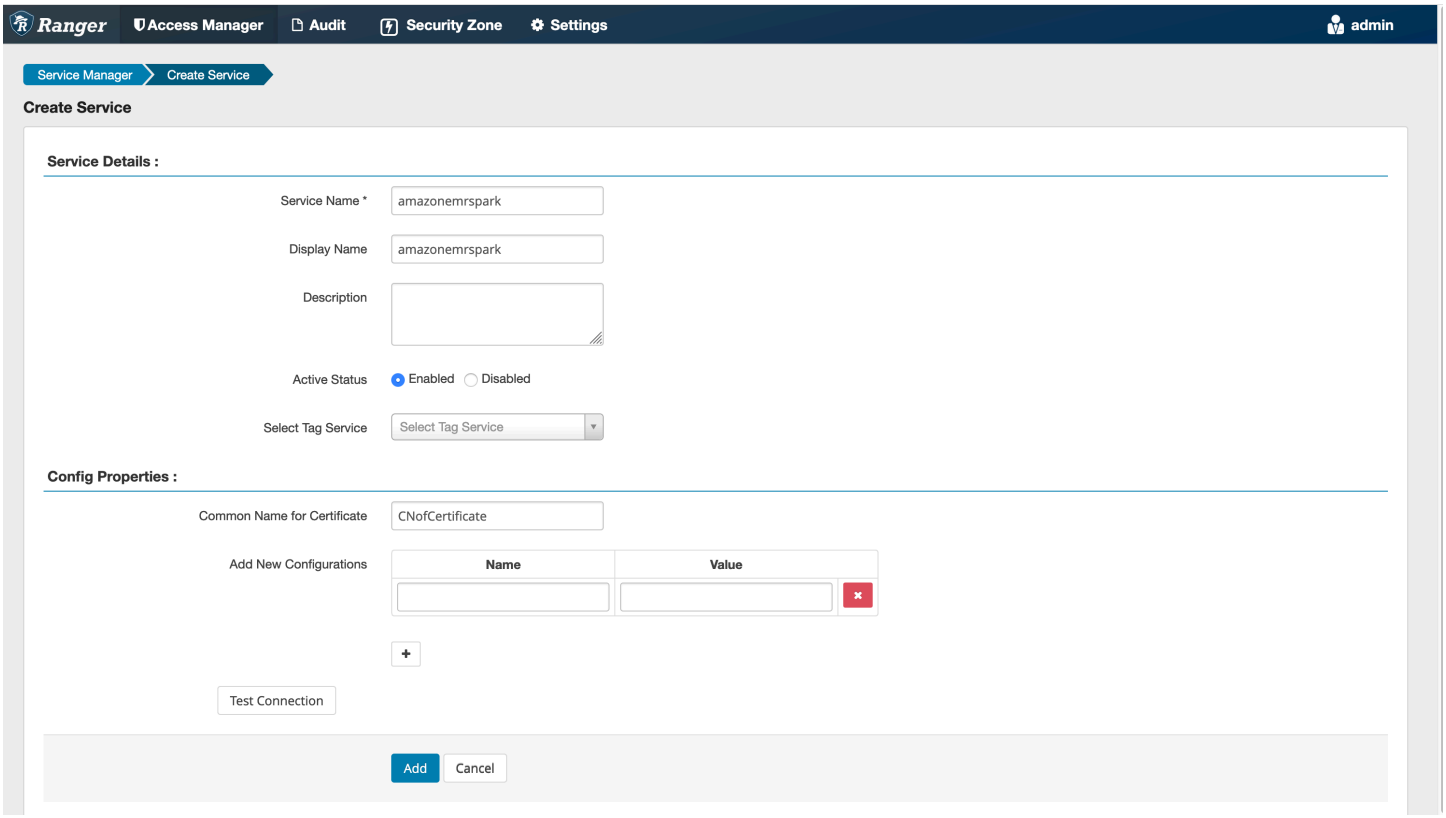


Étape 5 : créez une instance de l'application AMAZON-EMR-SPARK

Nom du service (s'il est affiché) : nom du service qui sera utilisé. La valeur suggérée est **`amazonemrspark`**. Notez le nom de ce service car il sera nécessaire lors de la création d'une configuration de sécurité EMR.

Nom d'affichage : nom à afficher pour cette instance. La valeur suggérée est **`amazonemrspark`**.

Nom commun du certificat : champ CN du certificat utilisé pour se connecter au serveur d'administration à partir d'un plug-in client. Cette valeur doit correspondre au champ CN de votre certificat TLS créé pour le plug-in.



> **Note**
>
> Le certificat TLS pour ce plug-in doit avoir été enregistré dans le magasin de confiance sur le serveur Ranger Admin. Pour plus d'informations, consultez Certificat TLS.

Création de politiques SparkSQL

Lors de la création d'une nouvelle politique, les champs à remplir sont les suivants :

Nom de politique : le nom de cette politique.

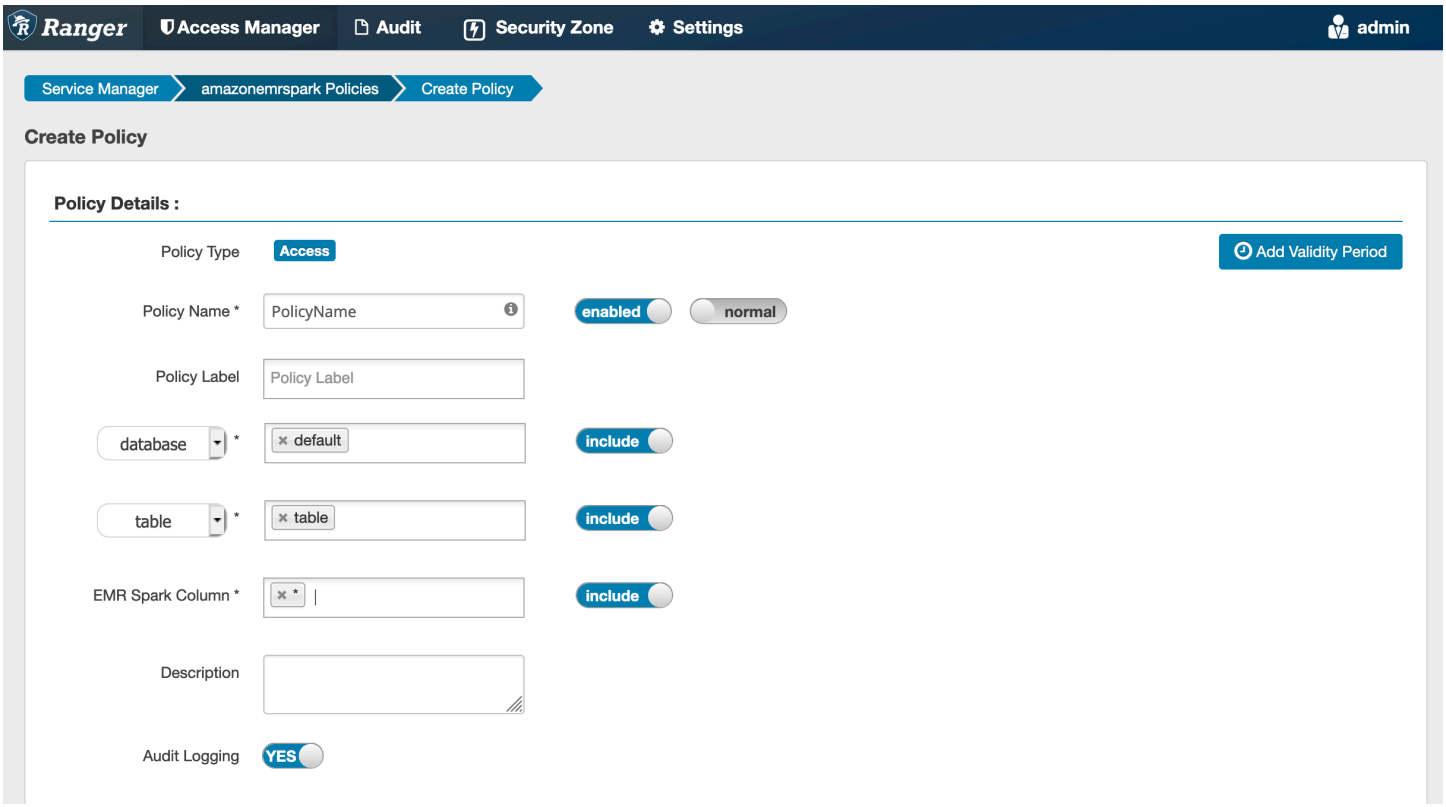
Étiquette de politique : une étiquette que vous pouvez attribuer à cette politique.

Base de données : base de données à laquelle cette politique s'applique. Le caractère générique « * » représente toutes les bases de données.

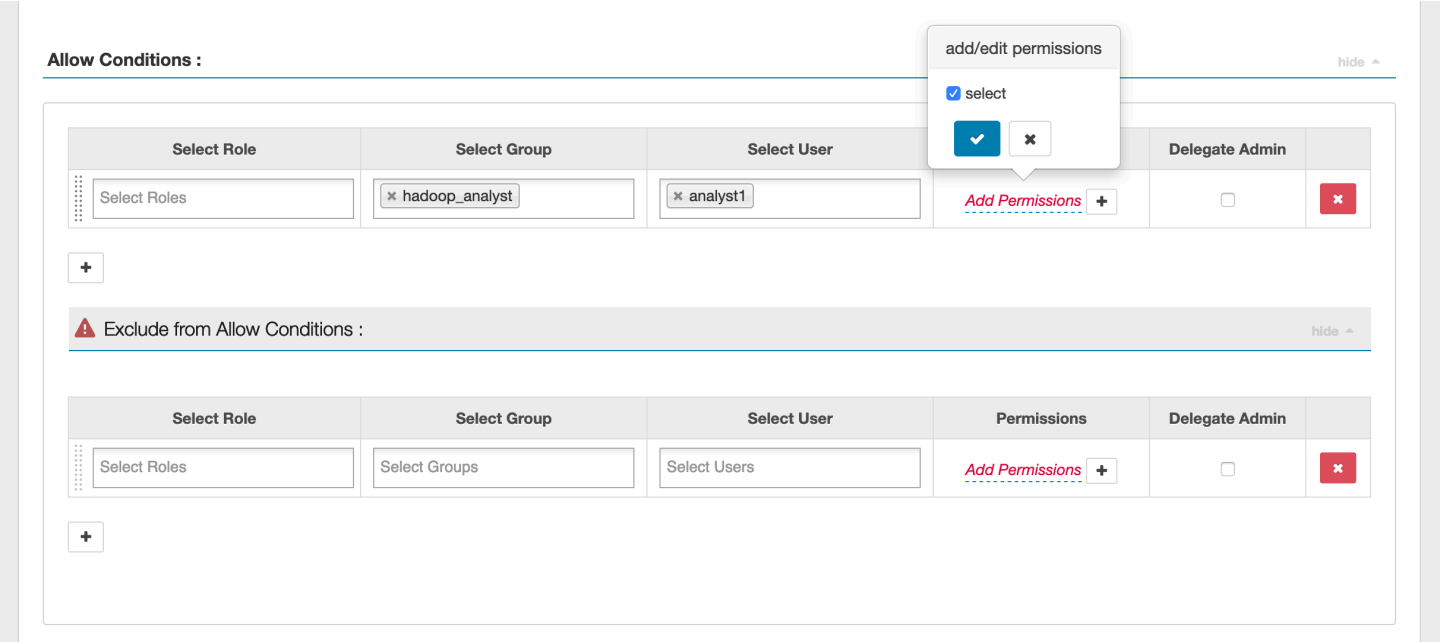
Table : les tables auxquelles cette politique s'applique. Le caractère générique "*" représente toutes les tables.

Colonne EMR Spark : colonnes auxquelles s'applique cette politique. Le caractère générique "*" représente toutes les colonnes.

Description : description de cette stratégie.



Pour spécifier les utilisateurs et les groupes, entrez les utilisateurs et les groupes ci-dessous pour accorder des autorisations. Vous pouvez également spécifier des exclusions pour les conditions autoriser et refuser.



Après avoir spécifié les conditions d'autorisation et de refus, cliquez sur Enregistrer.

Considérations

Chaque nœud du cluster EMR doit être en mesure de se connecter au nœud principal sur le port 9083.

Limites

Les limitations actuelles du plug-in Apache Spark sont les suivantes :

- Le serveur d'enregistrement se connecte toujours à HMS s'exécutant sur un cluster Amazon EMR. Configurez HMS pour qu'il se connecte au mode distant, si nécessaire. Vous ne devez pas placer de valeurs de configuration dans le fichier de configuration Hive-site.xml d'Apache Spark.
- Les tableaux créés à l'aide de sources de données Spark au format CSV ou Avro ne sont pas lisibles à l'aide de l'EMR. RecordServer Utilisez Hive pour créer et écrire des données, et lisez avec Record.
- Les tables Delta Lake et Hudi ne sont pas prises en charge.
- Les utilisateurs doivent avoir accès à la base de données par défaut. C'est une exigence pour Apache Spark.
- Le serveur Ranger Admin ne prend pas en charge l'auto-complétion.
- Le plug-in SparkSQL pour Amazon EMR ne prend pas en charge les filtres de lignes ni le masquage des données.

- Lorsque vous utilisez ALTER TABLE avec Spark SQL, l'emplacement d'une partition doit être le répertoire enfant d'un emplacement de table. L'insertion de données dans une partition dont l'emplacement est différent de celui de la table n'est pas prise en charge.

Plug-in EMRFS S3

Pour faciliter le contrôle d'accès aux objets dans S3 sur un cluster multi-tenant, le plug-in EMRFS S3 fournit des contrôles d'accès aux données de S3 lorsque vous y accédez via EMRFS. Vous pouvez autoriser l'accès aux ressources S3 au niveau des utilisateurs et des groupes.

Pour ce faire, lorsque votre application tente d'accéder à des données dans S3, EMRFS envoie une demande d'informations d'identification au processus de l'agent secret, où la demande est authentifiée et autorisée par le biais d'un plug-in Apache Ranger. Si la demande est autorisée, l'agent secret assume le rôle IAM pour les moteurs Apache Ranger avec une politique restreinte pour générer des informations d'identification qui n'ont accès qu'à la politique Ranger autorisant l'accès. Les informations d'identification sont ensuite renvoyées à EMRFS pour accéder à S3.

Rubriques

- Fonctionnalités prises en charge
- Installation de la configuration du service
- Création de politiques EMRFS S3
- Notes d'utilisation des politiques EMRFS S3
- Limites

Fonctionnalités prises en charge

Le plug-in EMRFS S3 fournit une autorisation au niveau de stockage. Des politiques peuvent être créées pour permettre aux utilisateurs et aux groupes d'accéder aux compartiments et aux préfixes S3. L'autorisation n'est accordée qu'à l'encontre d'EMRFS.

Installation de la configuration du service

Pour installer la définition du service EMRFS, vous devez configurer le serveur d'administration Ranger. Pour configurer le serveur, voirConfiguration du serveur d'administration Ranger.

Procédez comme suit pour installer la définition du service EMRFS.

Étape 1 : connectez-vous en SSH au serveur d'administration Apache Ranger.

Par exemple :

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Étape 2 : Téléchargez la définition du service EMRFS

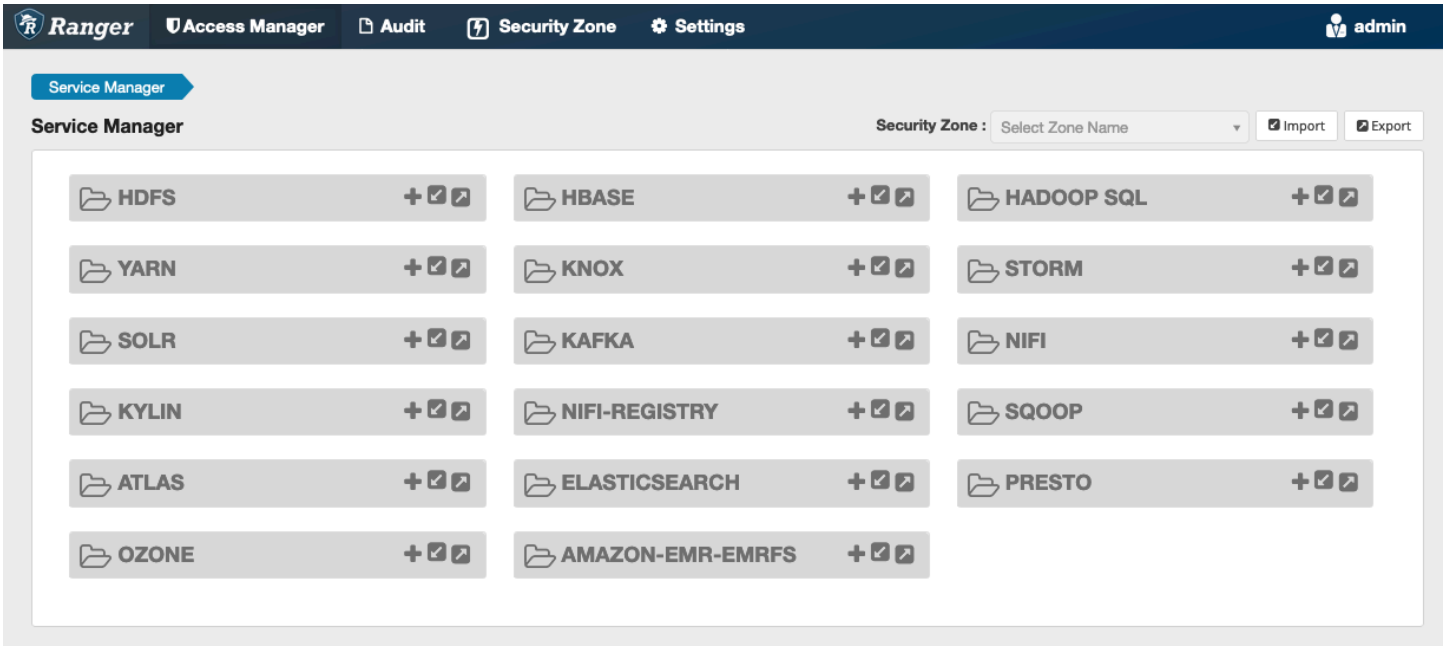
Dans un répertoire temporaire, téléchargez la définition du service Amazon EMR. Cette définition de service est prise en charge par les versions 2.x de Ranger.

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-emrfs.json
```

Étape 3 : enregistrer la définition du service EMRFS S3.

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-emrfs.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Si cette commande s'exécute correctement, vous verrez apparaître un nouveau service dans l'interface utilisateur d'administration de Ranger appelé « AMAZON-EMR-S3 », comme indiqué dans l'image suivante (la version 2.0 de Ranger est illustrée).



Étape 4 : Créez une instance de l'application AMAZON-EMR-EMRFS.

Créez une instance de la définition de service.

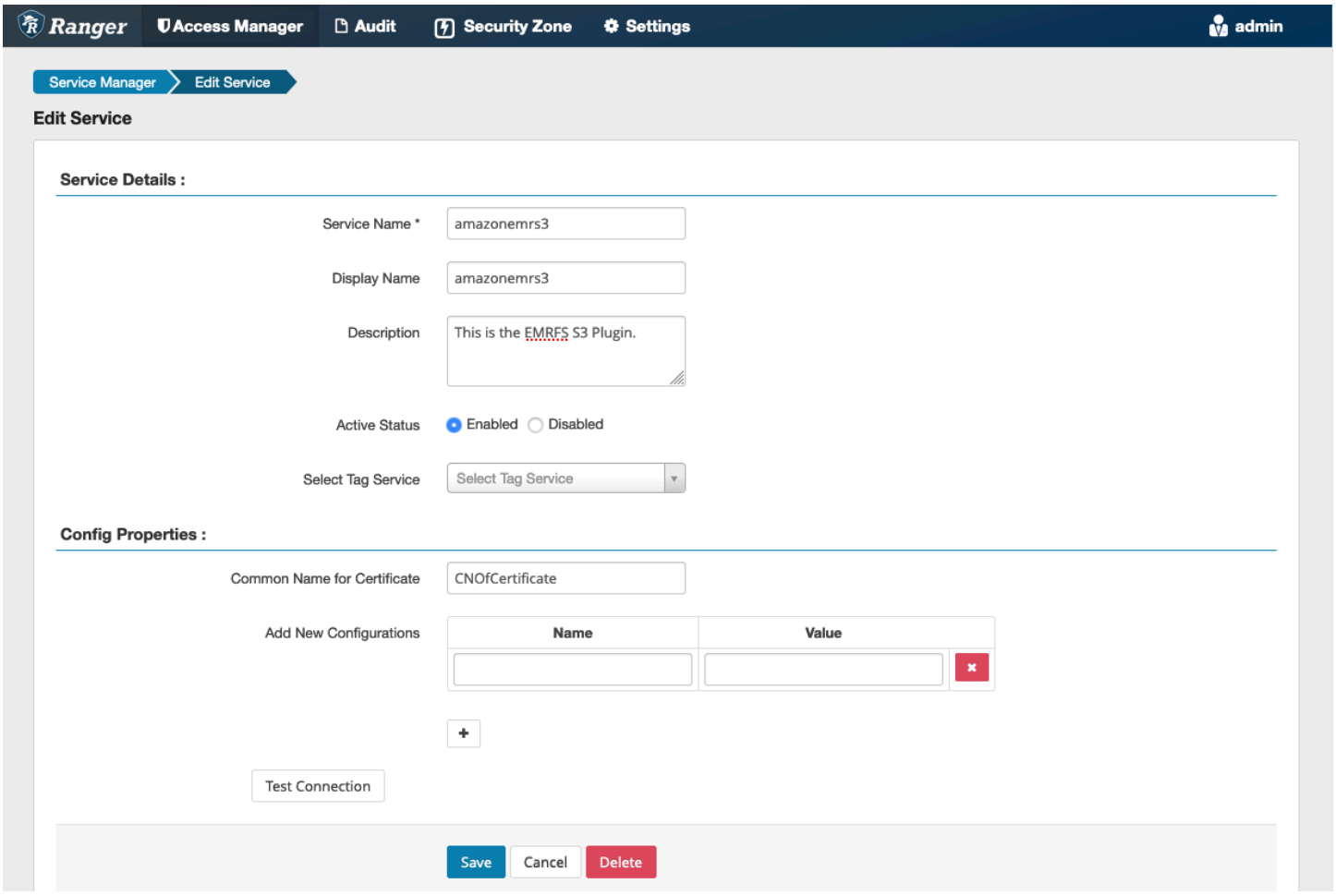
- Cliquez sur le + à côté de AMAZON-EMR-EMRFS.

Remplissez les champs suivants :

Nom du service (s'il est affiché) : la valeur suggérée est **`amazonemrspark`**. Notez le nom de ce service car il sera nécessaire lors de la création d'une configuration de sécurité EMR.

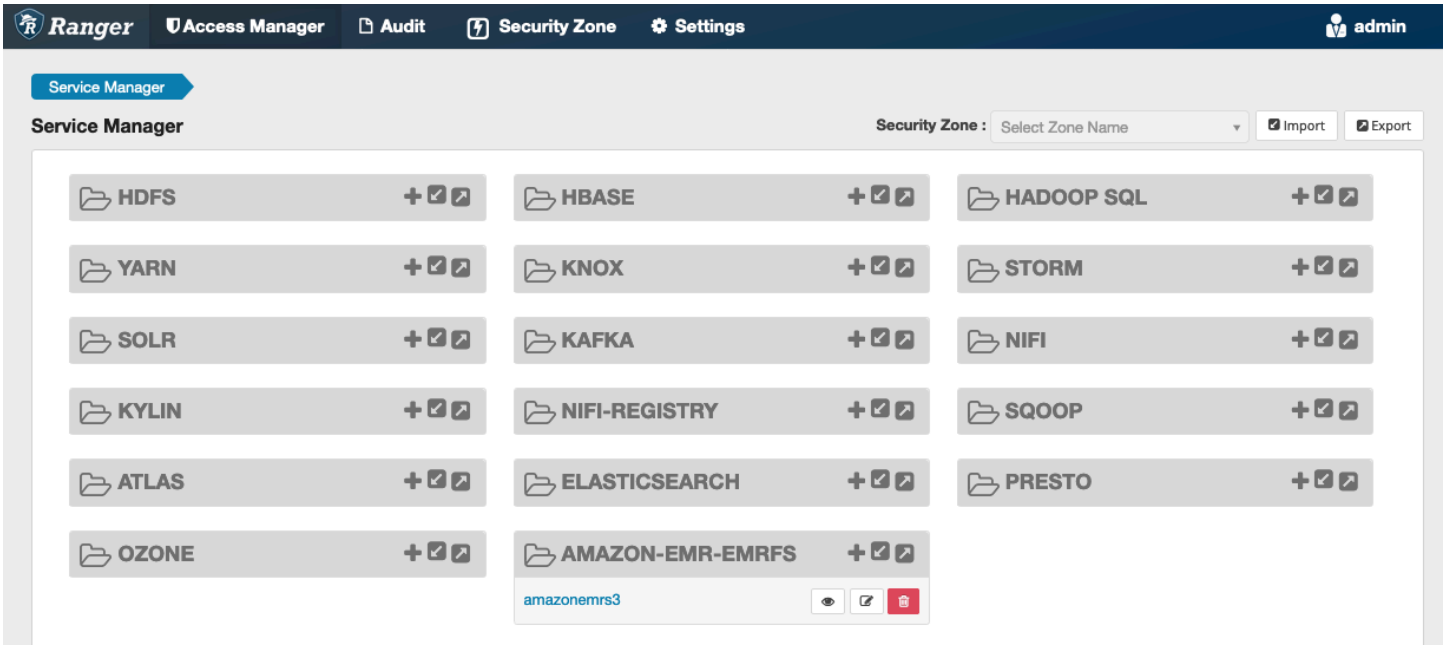
Nom d'affichage : nom affiché pour ce service. La valeur suggérée est **`amazonemrspark`**.

Nom commun du certificat : champ CN du certificat utilisé pour se connecter au serveur d'administration à partir d'un plug-in client. Cette valeur doit correspondre au champ CN du certificat TLS créé pour le plug-in.



> **Note**
>
> Le certificat TLS pour ce plug-in doit avoir été enregistré dans le magasin de confiance sur le serveur Ranger Admin. Pour plus d'informations, consultez Certificat TLS.

Lorsque le service est créé, le gestionnaire de services inclut « AMAZON-EMR-EMRFS », comme indiqué dans l'image suivante.



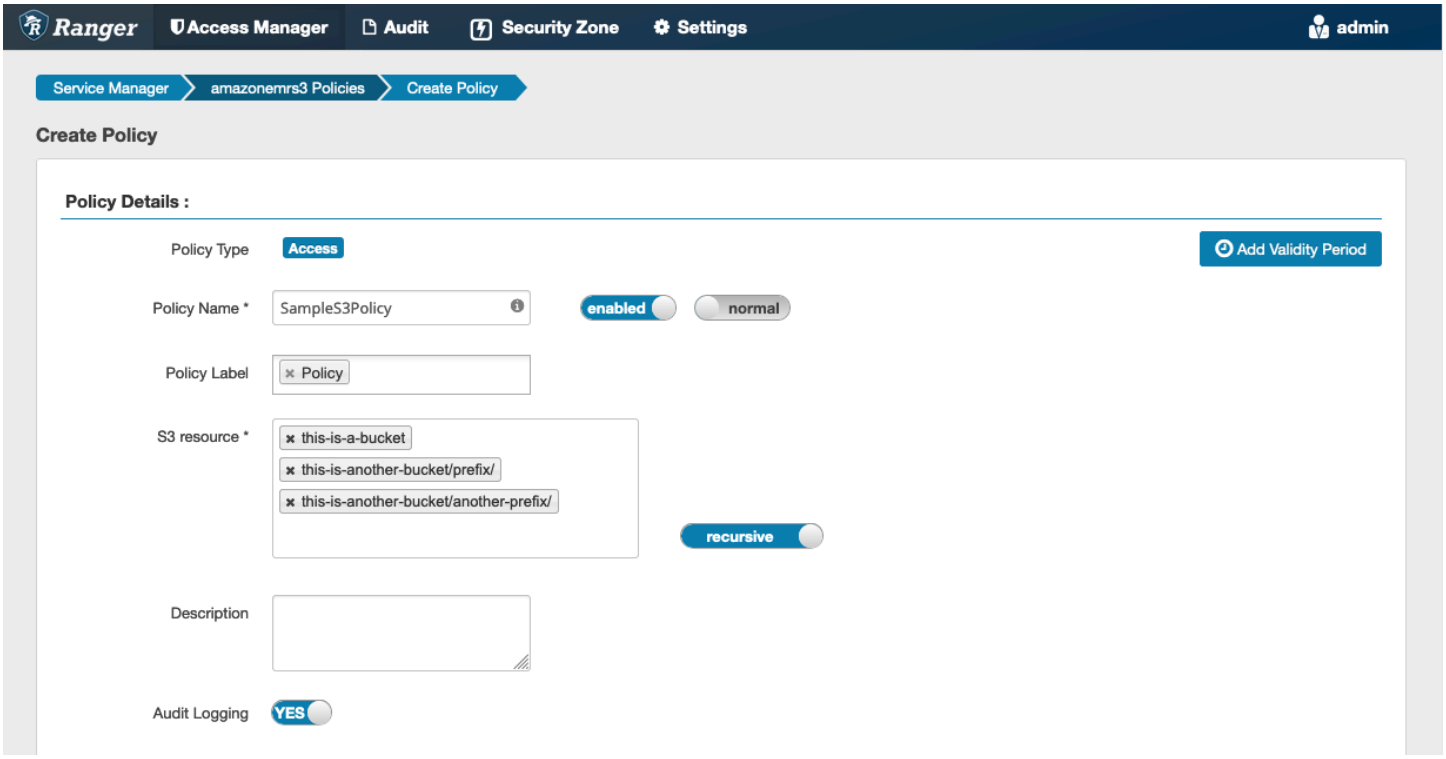
Création de politiques EMRFS S3

Pour créer une nouvelle politique sur la page Créer une politique du Service Manager, renseignez les champs suivants.

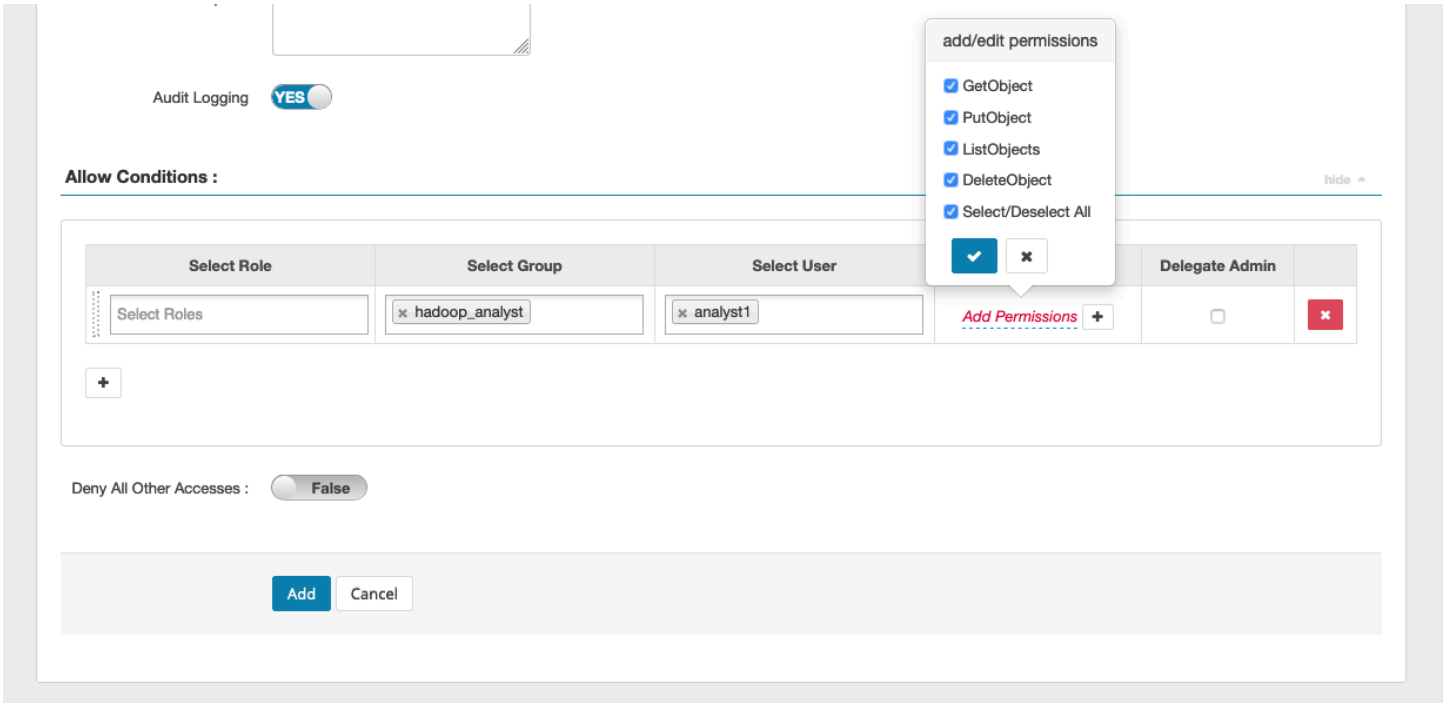
Nom de politique : le nom de cette politique.

Étiquette de politique : une étiquette que vous pouvez attribuer à cette politique.

Ressource S3 : ressource commençant par le compartiment et le préfixe facultatif. Consultez Notes d'utilisation des politiques EMRFS S3 pour obtenir des informations sur les bonnes pratiques. Les ressources du serveur d'administration Ranger ne doivent pas contenir **`s3://`**, **`s3a://`** ou **`s3n://`**.



Vous pouvez spécifier les utilisateurs et les groupes auxquels accorder des autorisations. Vous pouvez également spécifier des exclusions pour les conditions autoriser et refuser.



> **Note**
>
> Trois ressources au maximum sont autorisées pour chaque politique. L'ajout de plus de trois ressources peut entraîner une erreur lorsque cette politique est utilisée sur un cluster EMR. L'ajout de plus de trois politiques affiche un rappel concernant la limite de politique.

Notes d'utilisation des politiques EMRFS S3

Lors de la création de politiques S3 dans Apache Ranger, il convient de prendre en compte certaines considérations d'utilisation.

Autorisations d'accès à plusieurs objets S3

Vous pouvez utiliser des politiques récursives et des expressions génériques pour accorder des autorisations à plusieurs objets S3 dotés de préfixes communs. Les politiques récursives accordent des autorisations à tous les objets ayant un préfixe commun. Les expressions génériques sélectionnent plusieurs préfixes. Ensemble, ils donnent des autorisations à tous les objets ayant plusieurs préfixes communs, comme le montrent les exemples suivants.

Example Utilisation d'une politique récursive

Supposons que vous souhaitiez obtenir des autorisations pour répertorier tous les fichiers de parquet d'un compartiment S3 organisés comme suit.

```
s3://sales-reports/americas/
    +- year=2000
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    +- year=2019
    |      +- data-q1.json
    |      +- data-q2.json
    |      +- data-q3.json
    |      +- data-q4.json
    |
    +- year=2020
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    |      +- data-q3.parquet
    |      +- data-q4.parquet
    |      +- annual-summary.parquet
```

```
    +- year=2021
```

Tout d'abord, considérez les fichiers de parquet avec le préfixe `s3://sales-reports/americas/year=2000`. Vous pouvez leur accorder des GetObject autorisations de deux manières :

Utilisation de politiques non récursives : l'une des options consiste à utiliser deux politiques non récursives distinctes, l'une pour le répertoire et l'autre pour les fichiers.

La première politique autorise le préfixe `s3://sales-reports/americas/year=2020` (il n'y a pas de `/` à la fin).

```
- S3 resource = "sales-reports/americas/year=2000"
- permission = "GetObject"
- user = "analyst"
```

La deuxième politique utilise une expression générique pour accorder des autorisations à tous les fichiers avec un préfixe `sales-reports/americas/year=2020/` (notez le `/` à la fin).

```
- S3 resource = "sales-reports/americas/year=2020/*"
- permission = "GetObject"
- user = "analyst"
```

Utilisation d'une politique récursive : une alternative plus pratique consiste à utiliser une seule politique récursive et à accorder une autorisation récursive au préfixe.

```
 - S3 resource = "sales-reports/americas/year=2020"
 - permission = "GetObject"
 - user = "analyst"
 - is recursive = "True"
```

Jusqu'à présent, seuls les fichiers de parquet avec le préfixe `s3://sales-reports/americas/year=2000` ont été inclus. Vous pouvez désormais également inclure les fichiers parquet avec un préfixe différent, `s3://sales-reports/americas/year=2020`, dans la même politique récursive en introduisant une expression générique comme suit.

```
 - S3 resource = "sales-reports/americas/year=20?0"
 - permission = "GetObject"
 - user = "analyst"
 - is recursive = "True"
```

Politiques PutObject et DeleteObject autorisations

La rédaction de politiques `PutObject` et d'`DeleteObject`autorisations pour les fichiers sur EMRFS nécessite une attention particulière car, contrairement aux GetObject autorisations, elles nécessitent des autorisations récursives supplémentaires accordées au préfixe.

Example Politiques PutObject et DeleteObject autorisations

Par exemple, la suppression du fichier ne `annual-summary.parquet` nécessite pas seulement une DeleteObject autorisation d'accès au fichier lui-même.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "DeleteObject"
- user = "analyst"
```

Il nécessite également une politique accordant une récursivité `GetObject` et des autorisations `PutObject` à son préfixe.

De même, la modification du fichier `annual-summary.parquet` ne nécessite pas seulement une autorisation `PutObject` pour le fichier lui-même.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "PutObject"
- user = "analyst"
```

Il nécessite également une politique accordant une autorisation `GetObject` récursive à son préfixe.

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Des caractères génériques dans les politiques

Les caractères génériques peuvent être spécifiés dans deux domaines. Lorsque vous spécifiez une ressource S3, « * » et « ? » peut être utilisé. « * » fournit une correspondance avec un chemin S3 et correspond à tout ce qui se trouve après le préfixe. Par exemple, la politique suivante.

```
S3 resource = "sales-reports/americas/*"
```

Cela correspond aux chemins S3 suivants.

```
sales-reports/americas/year=2020/
sales-reports/americas/year=2019/
sales-reports/americas/year=2019/month=12/day=1/afile.parquet
sales-reports/americas/year=2018/month=6/day=1/afile.parquet
sales-reports/americas/year=2017/afile.parquet
```

Le caractère générique « ? » ne correspond qu'à un seul caractère. Par exemple, pour la politique.

```
S3 resource = "sales-reports/americas/year=201?/"
```

Cela correspond aux chemins S3 suivants.

```
sales-reports/americas/year=2019/
sales-reports/americas/year=2018/
sales-reports/americas/year=2017/
```

Caractères génériques dans les utilisateurs

Deux caractères génériques sont intégrés lors de l'attribution d'utilisateurs afin de leur donner accès. Le premier est le caractère générique est « {USER} » qui donne accès à tous les utilisateurs. Le deuxième caractère générique est « {OWNER} », qui donne accès au propriétaire d'un objet particulier ou directement. Cependant, le caractère générique « {USER} » n'est actuellement pas pris en charge.

Limites

Les limites actuelles du plug-in EMRFS S3 sont les suivantes :

- Les politiques d'Apache Ranger peuvent comporter au maximum trois politiques.
- L'accès à S3 doit être effectué via EMRFS et peut être utilisé avec des applications liées à Hadoop. Les éléments suivants ne sont pas pris en charge :

  - Bibliothèques Boto3

  - AWS SDK et CLI AWK

  - Connecteur open source S3A
- Les politiques de refus d'Apache Ranger ne sont pas prises en charge.
- Les opérations sur S3 avec des clés chiffrées par CSE-KMS ne sont actuellement pas prises en charge.

- La prise en charge interrégionale n'est pas prise en charge.
- La fonction de zone de sécurité dans Apache Ranger n'est pas prise en charge. Les restrictions de contrôle d'accès définies à l'aide de la fonctionnalité Zone de sécurité ne sont pas appliquées sur vos clusters Amazon EMR.
- L'utilisateur Hadoop ne génère aucun événement d'audit car Hadoop accède toujours au profil d'instance EC2.
- Il est recommandé de désactiver Amazon EMR Consistency View. Le S3 est très cohérent, il n'est donc plus nécessaire. Consultez Forte cohérence d'Amazon S3 pour plus d'informations.
- Le plug-in EMRFS S3 effectue de nombreux appels STS. Il est conseillé d'effectuer des tests de charge sur un compte de développement et de surveiller le volume d'appels STS. Il est également recommandé de faire une demande STS pour augmenter les limites AssumeRole de service.
- Le serveur d'administration Ranger ne prend pas en charge la saisie automatique.

Plug-in Trino

Trino (anciennement PrestoSQL) est un moteur de requêtes SQL que vous pouvez utiliser pour exécuter des requêtes sur des sources de données telles que HDFS, le stockage d'objets, les bases de données relationnelles et les bases de données NoSQL. Il élimine le besoin de migrer les données vers un emplacement central et vous permet d'interroger les données à tout moment. Amazon EMR fournit un plug-in Apache Ranger pour fournir des contrôles d'accès précis pour Trino. Le plug-in est compatible avec le serveur d'administration open source Apache Ranger version 2.0 et ultérieure.

Rubriques

- Fonctionnalités prises en charge
- Installation de la configuration du service
- Création de politiques Trino
- Considérations
- Limites

Fonctionnalités prises en charge

Le plug-in Apache Ranger pour Trino sur Amazon EMR prend en charge toutes les fonctionnalités du moteur de requête Trino, qui est protégé par un contrôle d'accès précis. Cela inclut les contrôles d'accès au niveau des bases de données, des tables et des colonnes, ainsi que le filtrage des lignes et le masquage des données. Les politiques Apache Ranger peuvent inclure des politiques d'octroi

et de refus pour les utilisateurs et les groupes. Les événements d'audit sont également soumis aux CloudWatch journaux.

Installation de la configuration du service

L'installation de la définition de service Trino nécessite que le serveur Ranger Admin soit configuré. Pour configurer le serveur Ranger Admin, consultez Configuration du serveur d'administration Ranger.

Procédez comme suit pour installer la définition de service Trino.

1. Connectez-vous en SSH au serveur d'administration Apache Ranger.

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

2. Désinstallez le plug-in du serveur Presto, s'il existe. Exécutez la commande suivante. Si le message d'erreur « Service introuvable » s'affiche, cela signifie que le plug-in du serveur Presto n'a pas été installé sur votre serveur. Passez à l'étape suivante.

```
curl -f -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin
 user_**_>_* -X DELETE -k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/
v2/api/servicedef/name/presto'
```

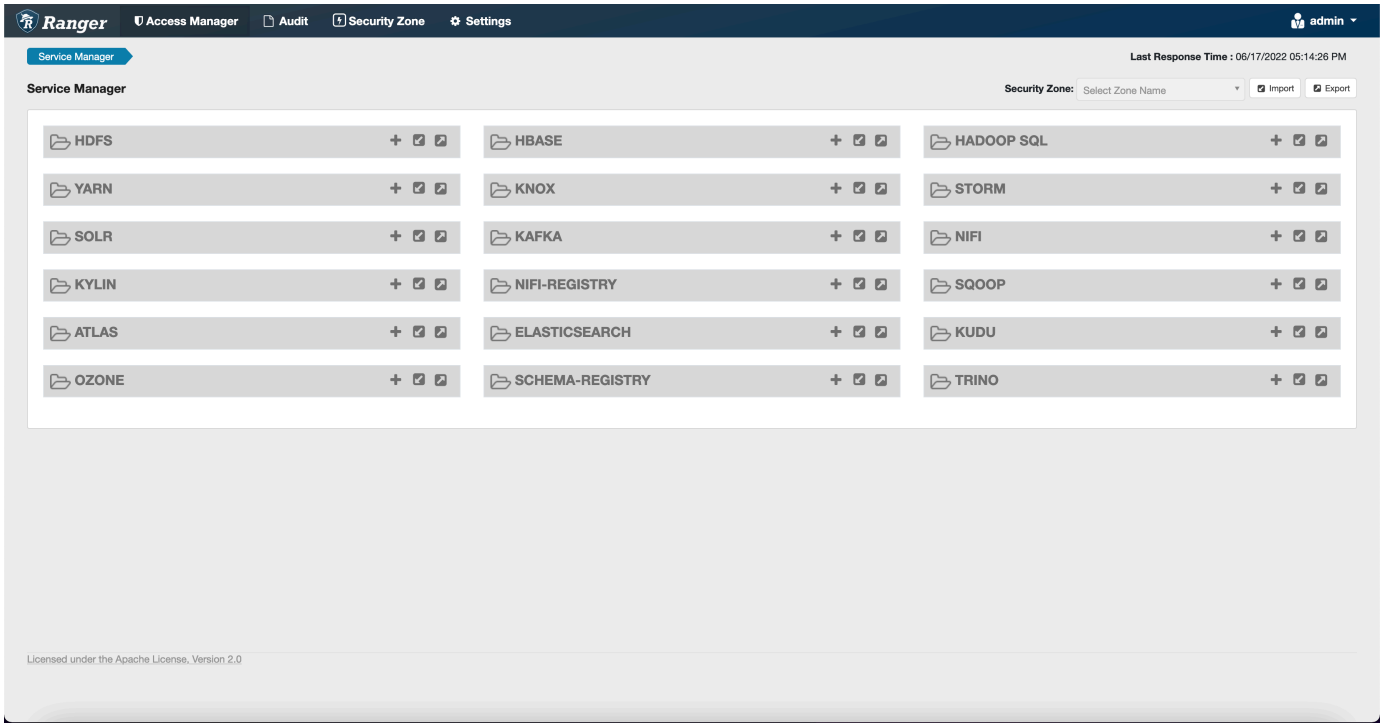
3. Téléchargez la définition du service et le plug-in du serveur d'administration Apache Ranger. Dans un répertoire temporaire, téléchargez la définition de service. Cette définition de service est prise en charge par les versions 2.x de Ranger.

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-trino.json
```

4. Enregistrez la définition du service Apache Trino pour Amazon EMR.

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_*
 -X POST -d @ranger-servicedef-amazon-emr-trino.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

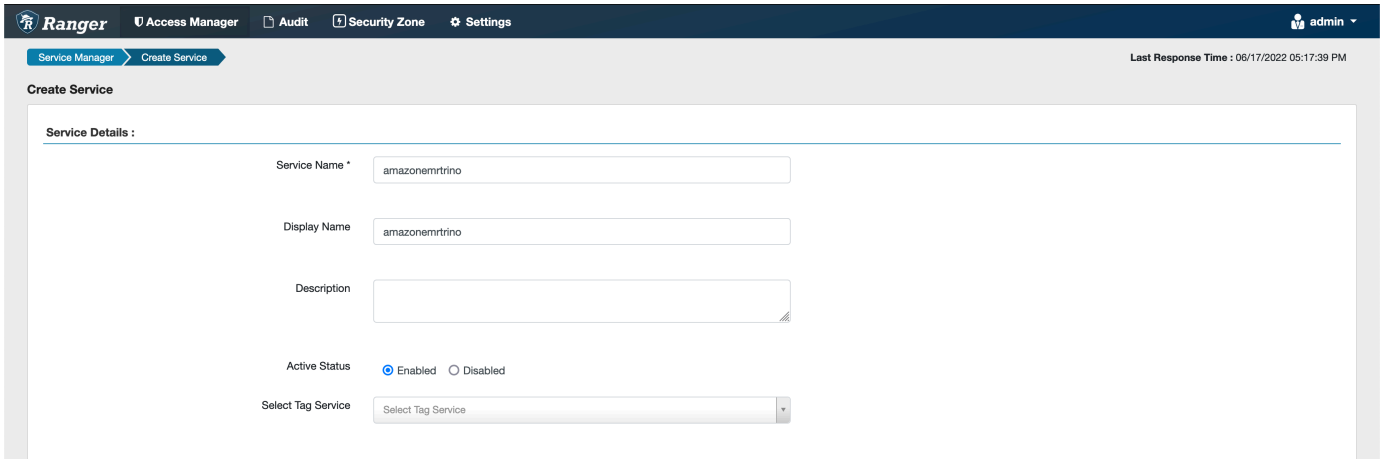
   Si cette commande s'exécute correctement, vous verrez un nouveau service appelé `TRINO` dans l'interface utilisateur de votre Ranger Admin, comme indiqué dans l'image suivante.



5. Créez une instance de l'application `TRINO` en saisissant les informations suivantes.

   Nom du service : nom du service que vous allez utiliser. La valeur suggérée est `amazonemrtrino`. Notez ce nom de service, car il sera nécessaire lors de la création d'une configuration de sécurité Amazon EMR.

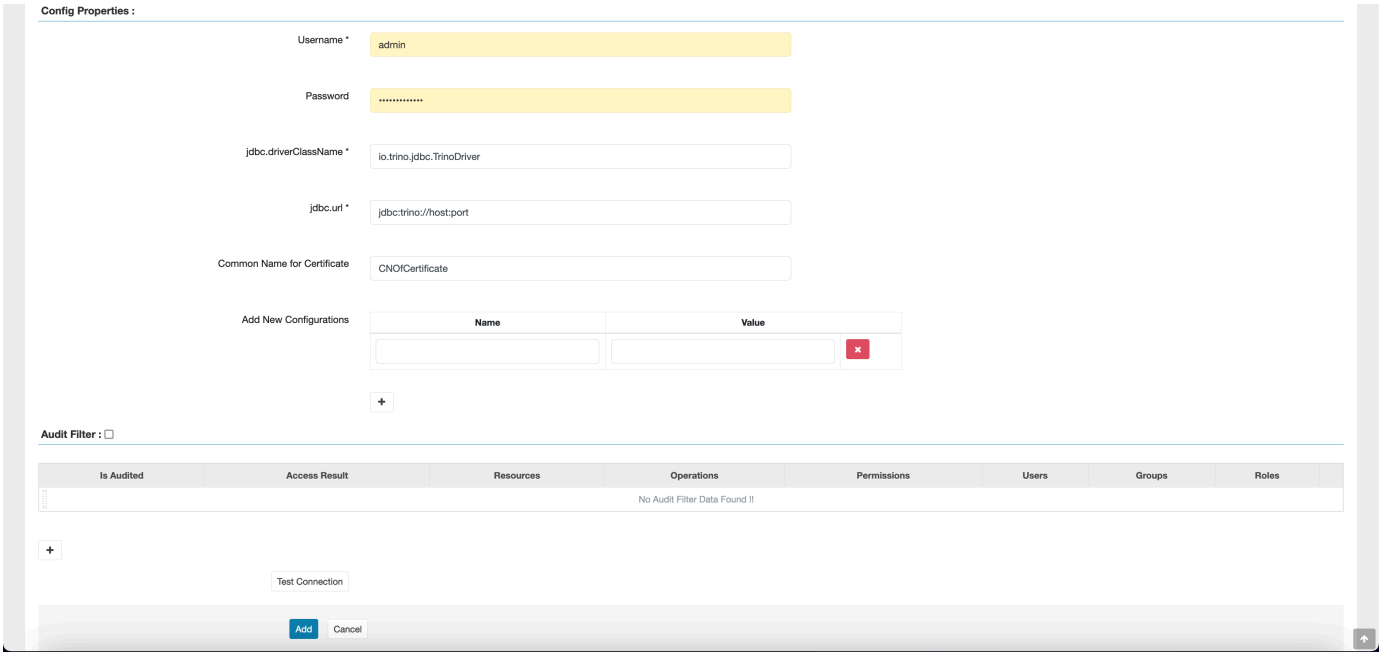
   Nom d'affichage : nom à afficher pour cette instance. La valeur suggérée est `amazonemrtrino`.



   jdbc.driver. ClassName: nom de classe de la classe JDBC pour la connectivité Trino. Vous pouvez utiliser la valeur par défaut.

   jdbc.url : chaîne de connexion JDBC à utiliser lors de la connexion au coordinateur Trino.

Nom commun du certificat : champ CN du certificat utilisé pour se connecter au serveur d'administration à partir d'un plug-in client. Cette valeur doit correspondre au champ CN de votre certificat TLS créé pour le plug-in.



Notez que le certificat TLS de ce plug-in doit avoir été enregistré dans le trust store du serveur d'administration Ranger. Pour plus d'informations, consultez Certificats TLS.

Création de politiques Trino

Lorsque vous créez une nouvelle politique, renseignez les champs suivants.

Nom de politique : le nom de cette politique.

Étiquette de politique : une étiquette que vous pouvez attribuer à cette politique.

Catalogue : catalogue auquel s'applique cette politique. Le caractère générique « * » représente tous les catalogues.

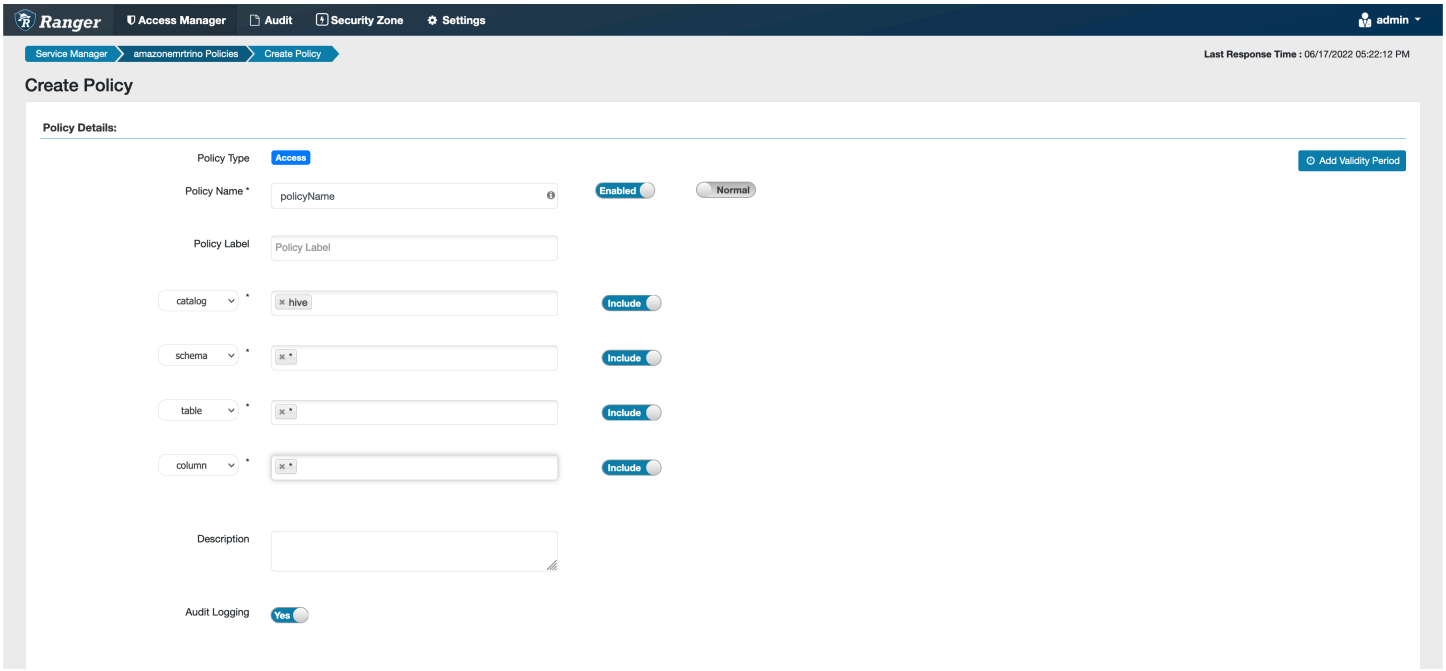
Schéma : schémas auxquels s'applique cette politique. Le caractère générique « * » représente tous les schémas.

Table : les tables auxquelles cette politique s'applique. Le caractère générique "*" représente toutes les tables.

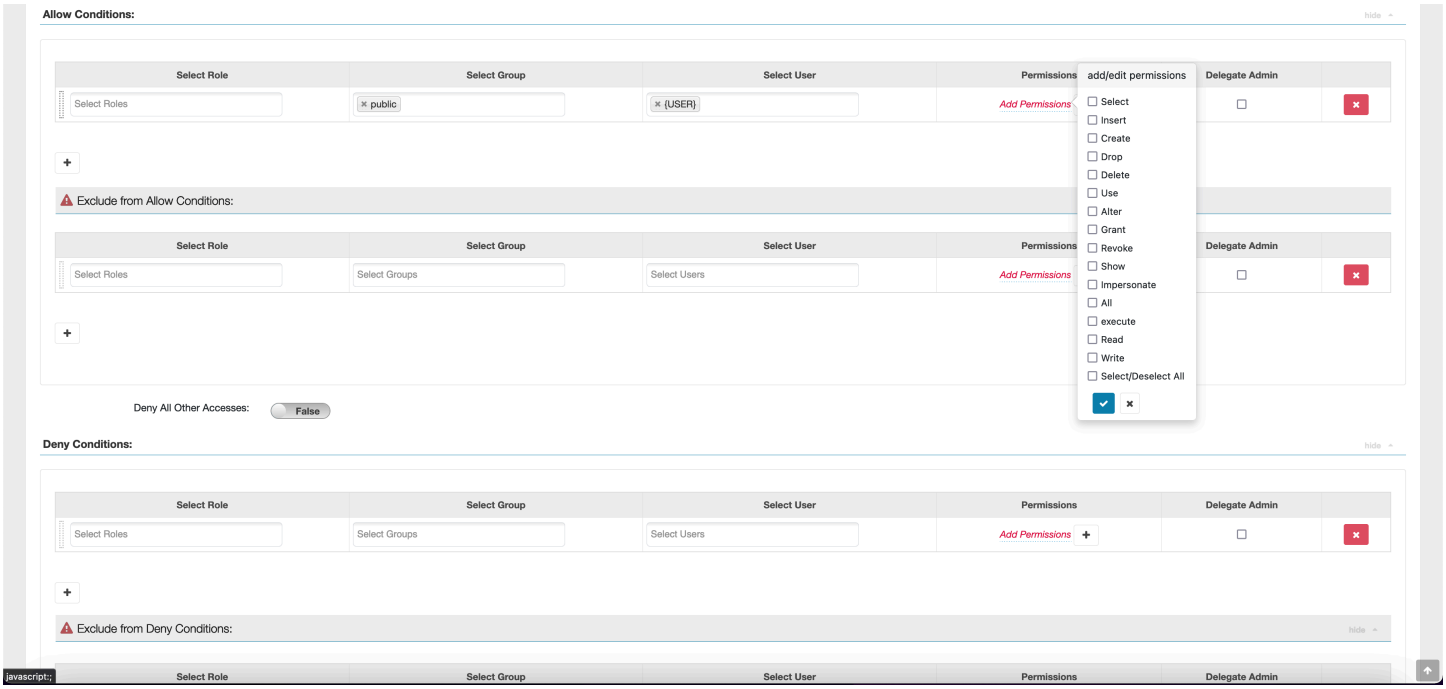
Colonne : colonnes auxquelles s'applique cette politique. Le caractère générique "*" représente toutes les colonnes.

Description : description de cette stratégie.

D'autres types de politiques existent pour l'Utilisateur Trino (pour l'accès par emprunt d'identité), la Propriété Trino System/Session (pour modifier les propriétés du système ou de la session du moteur), les Fonctions/procédures (pour autoriser les appels de fonction ou de procédure) et l'URL (pour accorder un accès en lecture/écriture au moteur sur les emplacements de données).



Pour accorder des autorisations à des utilisateurs et à des groupes spécifiques, entrez les utilisateurs et les groupes. Vous pouvez également spécifier des exclusions pour les conditions autoriser et refuser.



Après avoir spécifié les conditions d'autorisation et de refus, choisissez Enregistrer.

Considérations

Lorsque vous créez des politiques Trino dans Apache Ranger, vous devez tenir compte de certaines considérations d'utilisation.

Serveur de métadonnées Hive

Le serveur de métadonnées Hive n'est accessible que par des moteurs fiables, en particulier le moteur Trino, afin de se protéger contre tout accès non autorisé. Le serveur de métadonnées Hive est également accessible par tous les nœuds du cluster. Le port 9083 requis permet à tous les nœuds d'accéder au nœud principal.

Authentification

Par défaut, Trino est configuré pour s'authentifier à l'aide de Kerberos conformément à la configuration de sécurité Amazon EMR.

Chiffrement en transit requis

Le plug-in Trino nécessite que le chiffrement en transit soit activé dans la configuration de sécurité Amazon EMR. Pour activer le chiffrement, consultez Chiffrement en transit.

Limites

Les limites actuelles du plug-in Trino sont les suivantes :

- Le serveur Ranger Admin ne prend pas en charge l'auto-complétion.

Résolution des problèmes liés à Apache

Voici quelques problèmes fréquemment diagnostiqués liés à l'utilisation d'Apache Ranger.

Recommandations

- Test à l'aide d'un cluster à nœud principal unique : les clusters principaux à nœud unique sont fournis plus rapidement qu'un cluster à nœuds multiples, ce qui peut réduire le temps nécessaire à chaque itération de test.
- Définissez le mode de développement sur le cluster. Lorsque vous démarrez votre cluster EMR, définissez le paramètre `--additional-info"` sur :

  ```
  '{"clusterType":"development"}'
  ```

  Ce paramètre ne peut être défini que via la AWS CLI ou le AWS SDK et n'est pas disponible via la console Amazon EMR. Lorsque cet indicateur est activé et que le maître ne procède pas au provisionnement, le service Amazon EMR maintient le cluster en vie pendant un certain temps avant de le mettre hors service. Cette période est très utile pour analyser différents fichiers journaux avant la fermeture du cluster.

Le cluster EMR n'a pas pu être provisionné

Plusieurs raisons peuvent expliquer l'échec du démarrage d'un cluster Amazon EMR. Voici quelques méthodes pour diagnostiquer le problème.

Consulter les journaux de provisionnement EMR

Amazon EMR utilise Puppet pour installer et configurer des applications sur un cluster. L'examen des journaux permet de savoir si des erreurs se sont produites lors de la phase de provisionnement d'un cluster. Les journaux sont accessibles sur le cluster ou sur S3 si les journaux sont configurés pour être envoyés vers S3.

Les journaux sont stockés `/var/log/provision-node/apps-phase/0/{UUID}/puppet.log` sur le disque et `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/provision-node/apps-phase/0/{UUID}/puppet.log.gz`.

Messages d'erreur courants

Message d’erreur	Cause
Puppet (err) : échec du démarrage du système ! emr-record-server journal journalctl pour le serveur emr-record-server :	Le serveur d'enregistrement EMR n'a pas pu démarrer. Consultez les journaux du serveur d'enregistrement EMR ci-dessous.
Puppet (err) : échec du démarrage du système ! emr-record-server journal journalctl pour emrsecretagent :	EMR Secret Agent n'a pas pu démarrer. Consultez la section Vérifier les journaux de Secret Agent ci-dessous.
/Stage[main]/Ranger_plugins::Ranger_hive_plugin/Ranger_plugins::Prepare_two_way_tls[configure 2-way TLS in Hive plugin]/Exec[create keystore and truststore for Ranger Hive plugin]/returns (notice): 140408606197664:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:707:Expecting: ANY PRIVATE KEY	Le certificat TLS privé dans Secret Manager pour le certificat du plug-in Apache Ranger n'est pas au bon format ou n'est pas un certificat privé. Consultez Certificat TLS pour les formats de certificats.
/Stage [main] /Ranger_Plugins : :Ranger_S3_Plugin/Ranger_Plugins : :Prepare_TWO_WAY_TLS [configurer le TLS bidirectionnel dans le plugin Ranger s3] /Exec [créer un keystore et un truststore pour le plugin Ranger amazon-emr-s 3] /returns (notice) : Une erreur s'est produite (exception) lors de l'appel de l'opération : User : arn:aws:sts : :xxxxxxxxxxx:Assumed-role/EMR_EC2_ /I-XXXXXXXXXXXX n'est pas autorisé à exécuter :	Le rôle de profil d'instance EC2 ne dispose pas des autorisations appropriées pour récupérer les certificats TLS auprès de Secrets Agent.

Message d’erreur	Cause
secretsmanager : Value on resource : ARN:AWS:SecretsManager:US-East-1:XXXXXXXXX:secret : -XXXXX AccessDenied GetSecretValue DefaultRole GetSecret AdminServer	

Vérifiez les SecretAgent journaux

Les journaux de l'agent secret se trouvent dans `/emr/secretagent/log/` sur un nœud EMR ou dans le répertoire de S3 `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/secretagent/`.

Messages d'erreur courants

Message d’erreur	Cause
Exception dans le thread « main » com.amazonaws.services.securitytoken.model. AWSSecurityTokenServiceException: L'utilisateur : arn:aws:sts : :XXXXXXXXXXXX:assumed-role/EMR_EC2_ DefaultRole /i-XXXXXXXXXXXXX n'est pas autorisé à exécuter : sts : on resource : arn:aws:iam : :XXXXXXXXXXXX-XXXX-XXXX/* Role* (Service : ; Code de statut : 403 ; Code d'erreur : ; ID de demande : XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX ; Proxy : null) AssumeRole RangerPlugin DataAccess AWSSecurityTokenService AccessDenied	L'exception ci-dessus signifie que le rôle de profil d'instance EMR EC2 n'est pas autorisé à assumer le rôle Role. RangerPlugin DataAccess veuillez consulter Rôles IAM pour une intégration native avec Apache Ranger.
ERROR qtp54617902-149: Web App Exception Occurred	Vous pouvez ignorer ces erreurs.

Message d’erreur	Cause
javax.ws.rs. NotAllowedException : méthode HTTP 405 non autorisée	

Vérifiez les journaux du serveur d'enregistrement (pour SparkSQL)

Les journaux du serveur d'enregistrement EMR sont disponibles dans /var/log/emr-record-server/ sur un nœud EMR, ou ils se trouvent dans le répertoire s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/emr-record-server/ dans S3.

Messages d'erreur courants

Message d’erreur	Cause
InstanceMetadataServiceResourceFetcher:105 - [] Impossible de récupérer le jeton com.amazonaws. SdkClientException : échec de la connexion au point de terminaison du service	L'EMR SecretAgent ne s'est pas affiché ou présente un problème. Vérifiez la présence d'erreurs dans les SecretAgent journaux et dans le script de marionnette pour déterminer s'il y a eu des erreurs de provisionnement.

Les requêtes échouent de façon inattendue

Consultez les journaux du plugin Apache Ranger (journaux Apache HiveRecordServer, EMR, SecretAgent EMR, etc.)

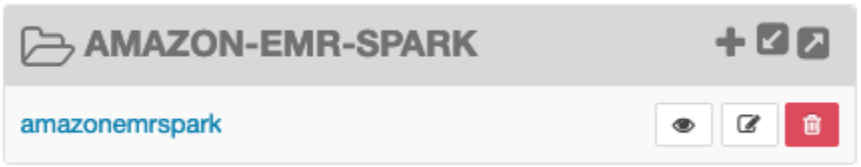
Cette section est commune à toutes les applications qui s'intègrent au plugin Ranger, telles que Apache Hive, EMR Record Server et EMR. SecretAgent

Messages d'erreur courants

Message d’erreur	Cause
ERREUR : 272 PolicyRefresher - [] (PolicyRefresherServiceName=Policy-Repository) : le	Ce message d'erreur signifie que le nom du service que vous avez fourni dans la configuration de sécurité EMR ne correspond pas à

Message d'erreur	Cause
service n'a pas été trouvé. Nettoiera le cache local des politiques (-1)	un référentiel de politiques de service sur le serveur d'administration Ranger.

Si, dans le serveur d'administration Ranger, votre service AMAZON-EMR-SPARK ressemble à ce qui suit, vous devez le saisir **`amazonemrspark`** comme nom du service.



Utilisation des vues du catalogue de données AWS Glue (aperçu)

> **Note**
>
> AWS Les vues du catalogue Glue Data dans Amazon EMR sont en version préliminaire et sont susceptibles d'être modifiées. La fonctionnalité est fournie en tant que service en version préliminaire tel que défini dans les Conditions de service AWS.

Vous pouvez créer et gérer des vues communes uniques dans le catalogue de données AWS Glue. Les vues communes uniques sont utiles car elles prennent en charge plusieurs moteurs de requêtes SQL. Vous pouvez donc accéder à la même vue sur différents moteurs Services AWS, tels qu'Amazon EMR, Amazon Athena et Amazon Redshift.

En créant une vue dans le catalogue de données, vous pouvez utiliser des autorisations de ressources et des contrôles d'accès basés sur des balises AWS Lake Formation pour accorder l'accès à une vue du catalogue de données. Avec cette méthode de contrôle d'accès, il n'est pas nécessaire de configurer un accès supplémentaire aux tables que vous avez référencées lors de la création de la vue. Cette méthode d'octroi d'autorisations est appelée sémantique du définisseur, et ces vues sont appelées vues du définisseur. Pour plus d'informations sur le contrôle d'accès dans Lake Formation, consultez la section Octroi et révocation d'autorisations sur les ressources du catalogue de données. dans le Guide AWS Lake Formation du développeur.

Les vues du catalogue de données sont utiles dans les cas d'utilisation suivants :

- Contrôle d'accès granulaire : créez une vue qui restreint l'accès aux données en fonction des autorisations dont l'utilisateur a besoin. Par exemple, vous pouvez utiliser des vues du catalogue de données pour empêcher les employés qui ne travaillent pas dans le service des ressources humaines (RH) de voir des données d'identification personnelle (PII).
- Définition complète de la vue : en appliquant certains filtres à votre vue dans le catalogue de données, vous vous assurez que les enregistrements de données contenus dans une vue du catalogue de données sont toujours complets.
- Sécurité renforcée : la définition de la requête utilisée pour créer la vue doit être complète. Cet avantage signifie que les vues du catalogue de données sont moins sensibles aux commandes SQL de joueurs malveillants.
- Partage de données simplifié : partagez des données avec d'autres Comptes AWS personnes sans déplacer aucune donnée. Pour plus d'informations, voir Partage de données entre comptes dans Lake Formation.

Création d'un affichage du Catalogue de données

> **Important**
>
> Dans cette version préliminaire, Amazon EMR ne valide pas le Spark-SQL que vous utilisez lors de la création de la vue. Pour réduire les risques, nous vous recommandons de limiter le nombre d'utilisateurs auxquels vous accordez des autorisations de création de vues.

Pour créer une vue du catalogue de données, vous devez utiliser un rôle IAM disposant des `SELECT` autorisations complètes avec des `Grantable` options sur toutes les tables auxquelles vous souhaitez faire référence lors de la création de la vue. Ce rôle est appelé rôle de définition. Pour obtenir la liste complète des autorisations et des conditions requises pour créer une vue du catalogue de données, consultez la section Utilisation des vues dans le guide du AWS Lake Formation développeur. Vous devez utiliser le AWS CLI pour configurer votre rôle IAM. Pour plus d'informations, consultez la section Utiliser un rôle IAM dans le AWS CLI.

Procédez comme suit pour créer une vue du catalogue de données.

> **Note**
>
> Pour accéder à une vue du catalogue de données depuis Apache Spark sur Amazon EMR, vous devez définir le dialecte to SPARK et le to. `DialectVersion 3.4.1-amzn-2`

1. Téléchargez d'abord le modèle d'aperçu.

```
aws s3 cp s3://emr-data-access-control-us-east-1/beta/glue-views/model/
service-2.json
```

2. Configurez le AWS CLI pour utiliser le modèle d'aperçu.

```
aws configure add-model --service-model file:///<path-to-preview-model>/
service-2.json --service-name glue-views
```

3. Créez la vue.

```
aws glue-views create-table --cli-input-json '{
  "DatabaseName": "<database>",
  "TableInput": {
    "Name": "<view>",
    "StorageDescriptor": {
      "Columns": [
        {
          "Name": "<col1>",
          "Type": "<data-type>"
        },
        ...
        {
          "Name": "<colN>",
          "Type": "<data-type>"
        }
      ]
    },
    "ViewDefinition": {
      "SubObjects": [
        "arn:aws:glue:<aws-region;>:<aws-account-id>:table/<database>/<referenced-
table1>",
        ...
        "arn:aws:glue:<aws-region>:<aws-account-id>:table/<database>/<referenced-
tableN>",
```

```
        ],
        "IsProtected": true,
       "Representations": [
         {
           "Dialect": "SPARK",
           "DialectVersion": "3.4.1-amzn-2",
           "ViewOriginalText": "<Spark-SQL>",
           "ViewExpandedText": "<Spark-SQL>"
         }
       ]
     }
   }
}'
```

Activation de l'accès à une vue du catalogue de données

> **Important**
>
> Nous vous recommandons d'activer l'accès aux vues du catalogue de données uniquement avec les clusters EMR dans les environnements de test et non dans les environnements de production.

Pour accéder à la vue du catalogue de données depuis Apache Spark sur Amazon EMR, vous devez d'abord activer la prise en charge de Lake Formation et utiliser le script ci-dessous pour activer la prise en charge des vues avec Spark sur Amazon EMR. Pour plus d'informations sur l'activation du support, consultez Enable Lake Formation with Amazon EMR et Use custom bootstrap actions.

```
# Download the script and upload it to Amazon S3
wget https://emr-data-access-control-us-east-1.s3.amazonaws.com/beta/glue-views/ba/
enable-mdv.sh /Users/$USER/enable-mdv.sh
aws s3 cp /Users/$USER/enable-views.sh s3://<bucket>/<prefix>/enable-views.sh

# EMR Security Configuration
cat <<EOT > /Users/$USER/lakeformation-protection.json
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
```

```
            "EnableApplicationScopedIAMRole":true
        },
        "LakeFormationConfiguration":{
            "AuthorizedSessionTagValue":"Amazon EMR"
        }
    },
     "EncryptionConfiguration": {
            "EnableInTransitEncryption": true,
            "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {
                 "CertificateProviderType": "PEM",
                 "S3Object": "s3://<BUCKET>/<PREFIX>/certificates.zip"
            }
        }
    }
}
EOT

SECURITY_CONFIG="RuntimeRolesWithAWSLakeFormation"

aws emr create-security-configuration \
--name $SECURITY_CONFIG \
--security-configuration file:///Users/$USER/lakeformation-protection.json

# EMR Cluster version
RELEASE_LABEL="emr-6.15.0"
```

Utilisez ensuite la AWS CLI commande suivante qui utilise l'action bootstrap pour créer un cluster EMR prenant en charge les vues du catalogue de données.

```
aws emr create-cluster \
...
--release-label $RELEASE_LABEL \
--security-configuration $SECURITY_CONFIG \
--bootstrap-actions \
 Name='Enable Views',Path="s3://<bucket>/<prefix>/enable-views.sh"
```

Interrogation d’un affichage du Catalogue de données

> **Important**
>
> Dans cette version préliminaire, nous vous recommandons d'accéder aux vues uniquement à partir de sources fiables. Dans la version préliminaire, Amazon EMR dispose d'un nombre limité de validations qui protègent votre cluster EMR.

Après avoir créé une vue du catalogue de données, vous pouvez désormais utiliser un rôle IAM pour interroger la vue. Le rôle IAM doit avoir l'`SELECT`autorisation d'accéder à la vue du catalogue de données. Il n'est pas nécessaire d'autoriser l'accès aux tables sous-jacentes référencées dans la vue. Vous devez utiliser ce rôle IAM comme rôle d'exécution. Vous pouvez accéder à la vue depuis un cluster EMR à l'aide d'un rôle d'exécution dans Amazon EMR Steps, EMR Studio et Studio. SageMaker Pour plus d'informations sur les rôles d'exécution, consultez la section Rôles d'exécution pour les étapes d'Amazon EMR.

Une fois que tout est configuré, vous pouvez demander votre avis. Par exemple, après avoir attaché le cluster EMR à votre espace de travail dans EMR Studio, vous pouvez exécuter la requête suivante pour accéder à une vue.

```
SELECT * from <database>.<glue-data-catalog-view> LIMIT 10
```

Limites

Tenez compte des limites suivantes lorsque vous utilisez les vues du catalogue de données.

- Vous ne pouvez créer des vues de catalogue de données qu'avec Amazon EMR 6.15.0.
- Vous ne pouvez référencer que 10 tables au maximum dans la définition de la vue.
- Vous ne pouvez créer que des vues `PROTECTED` de catalogue de données. `UNPROTECTED`les vues ne sont pas prises en charge.
- Vous ne pouvez pas référencer des tables Compte AWS dans une autre vue du catalogue de données.
- Les fonctions définies par l'utilisateur (UDF) ne sont pas prises en charge.
- Vous ne pouvez pas faire référence à des formats de table ouverte tels qu'Apache Hudi ou Apache Iceberg dans les vues du catalogue de données.

- Vous ne pouvez pas référencer d'autres vues dans les vues du catalogue de données.

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, veuillez consulter la section Amazon EC2 security groups for Linux instances (français non garanti) dans le Guide de l'utilisateur Amazon EC2.

Vous utilisez deux classes de groupes de sécurité avec 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 créés par Amazon EMR ou spécifier des groupes de sécurité gérés personnalisés. Quoi qu'il en soit, Amazon EMR ajoute automatiquement aux groupes de sécurité gérés des règles dont un cluster a besoin pour communiquer entre les instances du cluster et AWS les services.

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.

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.

> **Important**
>
> Soyez vigilant lorsque vous modifiez les règles du groupe de sécurité pour ouvrir des ports. Assurez-vous d'ajouter des règles qui autorisent uniquement le trafic provenant de clients approuvés et authentifiés pour les protocoles et les ports nécessaires à l'exécution de vos charges de travail.

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. Pour les AWS comptes créés après juillet 2019, le blocage de l'accès public à Amazon EMR est activé par défaut. Pour les AWS comptes ayant créé un cluster avant juillet 2019, le blocage de l'accès public par Amazon EMR est désactivé par défaut. Pour plus d'informations, consultez Utilisation du blocage de l'accès public Amazon EMR.

Rubriques

- Utilisation de groupes de sécurité gérés par Amazon EMR
- Utilisation des groupes de sécurité supplémentaires
- Spécification des groupes de sécurité gérés par Amazon EMR et des groupes de sécurité supplémentaires
- Spécification des groupes de sécurité EC2 pour les blocs-notes EMR
- Utilisation du blocage de l'accès public Amazon EMR

> **Note**
>
> Amazon EMR essaie d'utiliser des alternatives inclusives pour les termes industriels potentiellement offensants ou non inclusifs tels que « maître » et « esclave ». Nous avons adopté une nouvelle terminologie pour favoriser une expérience plus inclusive et faciliter votre compréhension des composants de service.
> Nous décrivons désormais les « nœuds » comme des instances, et les types d'instances Amazon EMR comme des instances primaires, de noyau et de tâches. Pendant la transition, il se peut que vous trouviez encore des références à des termes obsolètes, tels que ceux qui se rapportent aux groupes de sécurité pour Amazon EMR.

Utilisation de groupes de sécurité gérés par Amazon EMR

> **Note**
>
> Amazon EMR essaie d'utiliser des alternatives inclusives pour les termes industriels potentiellement offensants ou non inclusifs tels que « maître » et « esclave ». Nous avons adopté une nouvelle terminologie pour favoriser une expérience plus inclusive et faciliter votre compréhension des composants de service.
> Nous décrivons désormais les « nœuds » comme des instances, et les types d'instances Amazon EMR comme des instances primaires, de noyau et de tâches. Pendant la transition, il se peut que vous trouviez encore des références à des termes obsolètes, tels que ceux qui se rapportent aux groupes de sécurité pour 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 d'Amazon VPC.

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. De plus, 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'existent pas encore.

Vous pouvez modifier les règles dans les groupes de sécurité gérés après que les clusters soient créés. Lorsque vous créez un nouveau cluster, Amazon EMR vérifie les règles des groupes de sécurité gérés que vous spécifiez, puis crée toutes les règles entrantes manquantes 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 Amazon EMR par défaut est également ajoutée aux groupes de sécurité gérés par Amazon EMR personnalisés que vous spécifiez.

Les groupes de sécurité gérés par défaut sont les suivants :

- ElasticMapRéduire - primaire

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux publics).
- ElasticMapRéduire le noyau

  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).
- ElasticMapRéduis-Primaire-Privé

  Pour les règles de ce groupe de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux privés).
- ElasticMapReduce-Core-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).
- ElasticMapRéduisez- 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).

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 porte le nom de groupe ElasticMap Reduce-primary. Il est régi par les règles suivantes. Si vous spécifiez un groupe de sécurité géré personnalisé, Amazon EMR ajoute les mêmes règles à votre groupe de sécurité personnalisé.

Type	Protoc	Plage de ports	Source	Détails
Règles entrantes				
Tous les ICMP-IPv4	Tous	N/A	L'ID de groupe du groupe de sécurité géré pour l'instanc e 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-primary` 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	Tous		
Tous les UDP	UDP	Tous		
Tous les ICMP-IPV4	Tous	N/A	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	Tous		
Tous les UDP	UDP	Tous		

Type	Protoc	Plage de ports	Source	Détails
Personnalisé	TCP	8443	Différentes plages d'adresses IP Amazon	Ces règles permettent au gestionnaire de cluster de communiquer avec le nœud primaire.

Pour accorder à des sources fiables un accès SSH au groupe de sécurité principal via la console

Pour modifier vos groupes de sécurité, vous devez avoir l'autorisation de gérer les groupes de sécurité pour le VPC dans lequel se trouve le cluster. Pour plus d'informations, consultez Modification des autorisations d'un utilisateur et l'exemple de politique permettant de gérer les groupes de sécurité EC2 dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Clusters. Choisissez l'ID du cluster que vous souhaitez modifier.
3. Dans le volet Réseau et sécurité, développez la liste déroulante des groupes de sécurité EC2 (pare-feu).
4. Sous Nœud principal, choisissez votre groupe de sécurité.
5. Choisissez Modifier les règles entrantes.
6. Vérifiez s'il existe une règle entrante qui autorise l'accès public avec les paramètres suivants. Si elle existe, choisissez Supprimer pour la supprimer.

   - Type

     SSH
   - Port

     22
   - Source

     Personnalisé 0.0.0.0/0

> **Warning**
>
> Avant décembre 2020, une règle préconfigurée autorisait le trafic entrant sur le port 22 en provenance de toutes les sources. Cette règle a été créée pour simplifier les connexions SSH initiales au nœud primaire. Nous vous recommandons vivement de supprimer cette règle d'entrée et de limiter le trafic aux sources fiables.

7. Faites défiler la liste des règles jusqu'en bas et sélectionnez Ajouter une règle.
8. Dans le champ Type, sélectionnez SSH.

   En sélectionnant SSH, vous saisissez automatiquement TCP pour le protocole et 22 pour la plage de ports.
9. Pour source, sélectionnez Mon adresse IP pour ajouter automatiquement votre adresse IP en tant qu'adresse source. Vous pouvez également ajouter une plage d'adresses IP de clients fiables personnalisées ou créer des règles supplémentaires pour d'autres clients. De nombreux environnements réseau allouent des adresses IP de manière dynamique. Il se peut donc que vous deviez mettre à jour vos adresses IP pour les clients fiables à l'avenir.
10. Choisissez Enregistrer.
11. Choisissez éventuellement l'autre groupe de sécurité sous Nœuds principaux et de tâches dans le volet Réseau et sécurité et répétez les étapes ci-dessus pour autoriser l'accès du client SSH aux nœuds principaux et aux nœuds de tâches.

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 porte le nom de groupe ElasticMap Reduce-core. 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.

Type	Protoc	Plage de ports	Source	Détails
Règles entrantes				
Tous les ICMP-IPV4	Tous	N/A	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-core` 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	Tous		
Tous les UDP	UDP	Tous		
Tous les ICMP-IPV4	Tous	N/A	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	Tous		
Tous les UDP	UDP	Tous		

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 porte le nom de groupe ElasticMap Reduce-Primary-Private. 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.

Type	Protoc	Plage de ports	Source	Détails
Règles entrantes				
Tous les ICMP-IPv4	Tous	N/A	L'ID de groupe du groupe de sécurité géré pour l'instanc e 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-P rimary-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	Tous		
Tous les UDP	UDP	Tous		
Tous les ICMP-IPV4	Tous	N/A	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 atteignab les à partir du sous-réseau privé, même si les instances se trouvent dans des clusters différents.
Tous les TCP	TCP	Tous		
Tous les UDP	UDP	Tous		
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 primaire.
Règles sortantes				

Type	Protoc	Plage de ports	Source	Détails
Tout le trafic	Tous	Tous	0.0.0.0/0	Fournit un accès sortant à Internet.
TCP personnalisé	TCP	9443	L'ID de groupe du groupe de sécurité géré pour l'accès au service dans un sous-réseau privé.	Si la règle sortante par défaut « Tout le trafic » ci-dessus est supprimée, cette règle constitue une exigence minimale pour Amazon EMR 5.30.0 et versions ultérieures. **Note** Amazon EMR n'ajoute pas cette règle lorsque vous utilisez un groupe de sécurité géré personnalisé.
TCP personnalisé	TCP	80 (http) ou 443 (https)	L'ID de groupe du groupe de sécurité géré pour l'accès au service dans un sous-réseau privé.	Si la règle de sortie par défaut "All traffic" (tout le trafic) ci-dessus est supprimée, cette règle est une exigence minimale pour Amazon EMR 5.30.0 et les versions ultérieures afin de se connecter à Amazon S3 via https. **Note** Amazon EMR n'ajoute pas cette règle lorsque vous utilisez un groupe de sécurité géré personnalisé.

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 porte le nom de groupe ElasticMap Reduce-Core-Private. 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.

Type	Protoc	Plage de ports	Source	Détails
Règles entrantes				
Tous les ICMP-IPV4	Tous	N/A	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-core` 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	Tous		
Tous les UDP	UDP	Tous		
Tous les ICMP-IPV4	Tous	N/A	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	Tous		
Tous les UDP	UDP	Tous		
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.
Règles sortantes				

Type	Protoc	Plage de ports	Source	Détails
Tout le trafic	Tous	Tous	0.0.0.0/0	Reportez-vous à Modification des règles de sortie ci-dessous.
TCP personnalisé	TCP	80 (http) ou 443 (https)	L'ID de groupe du groupe de sécurité géré pour l'accès au service dans un sous-réseau privé.	Si la règle de sortie par défaut "All traffic" (tout le trafic) ci-dessus est supprimée, cette règle est une exigence minimale pour Amazon EMR 5.30.0 et les versions ultérieures afin de se connecter à Amazon S3 via https. **Note** Amazon EMR n'ajoute pas cette règle lorsque vous utilisez un groupe de sécurité géré personnalisé.

Modification des règles de sortie

Par défaut, Amazon EMR crée ce groupe de sécurité avec des règles de trafic sortant qui autorisent tout le trafic sortant sur tous les protocoles et ports. L'option Autoriser tout le trafic sortant est sélectionnée car les différentes applications Amazon EMR et clients pouvant s'exécuter sur des clusters Amazon EMR peuvent nécessiter des règles de sortie différentes. Amazon EMR n'est pas en mesure d'anticiper ces paramètres spécifiques lors de la création de groupes de sécurité par défaut. Vous pouvez délimiter la sortie dans vos groupes de sécurité afin d'inclure uniquement les règles adaptées à vos cas d'utilisation et à vos politiques de sécurité. Ce groupe de sécurité requiert au minimum les règles sortantes suivantes, mais certaines applications peuvent avoir besoin d'une sortie supplémentaire.

Type	Protoc	Plage de ports	Destination	Détails
Tous les TCP	TCP	Tous	pl-*xxxxxxxx*	Liste de préfixes Amazon S3 gérée `com.amazonaws.` *MyRegion*`.s3`.
Tout le trafic	Tous	Tous	sg-*xxxxxxxxx xxxxxxxx*	ID du groupe de sécurité `ElasticMapReduce-Core-Private` .
Tout le trafic	Tous	Tous	sg-*xxxxxxxxx xxxxxxxx*	ID du groupe de sécurité `ElasticMapReduce-Primary-Private` .
TCP personnalisé	TCP	9443	sg-*xxxxxxxxx xxxxxxxx*	ID du groupe de sécurité `ElasticMapReduce-ServiceAccess` .

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 aux services dans les sous-réseaux privés porte le nom de groupe ElasticMap Reduce- ServiceAccess. 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 primaire, ainsi qu'avec les nœuds de base et de tâche. Les mêmes règles sont nécessaires si vous utilisez des groupes de sécurité personnalisés.

Type	Protoc	Plage de ports	Source	Détails
Règles entrantes Requises pour les clusters Amazon EMR avec Amazon EMR version 5.30.0 et ultérieures.				
TCP personnalisé	TCP	9443	L'ID de groupe du groupe de sécurité	

Type	Protoc	Plage de ports	Source	Détails
			géré pour l'instance principale.	Cette règle permet la communication entre le groupe de sécurité de l'instance principale et le groupe de sécurité d'accès au service.
Règles sortantes Requises pour tous les clusters Amazon EMR				
TCP personnalisé	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 primaire, ainsi qu'avec les nœuds de base et de tâche.
TCP personnalisé	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 primaire, ainsi qu'avec les nœuds de base et de tâche.

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 15 groupes de sécurité supplémentaires pour l'instance principale, 15 pour les instances principales et de tâches, et 15 pour l'accès aux services (dans les 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 Limites des groupes de sécurité dans le Guide de l'utilisateur Amazon VPC.

Spécification des groupes de sécurité gérés par Amazon EMR et des groupes de sécurité supplémentaires

Vous pouvez spécifier des groupes de sécurité à l'aide de l'API AWS Management Console AWS CLI, de, ou de l'API Amazon 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).

New console

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

Pour spécifier les groupes de sécurité avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Mise en réseau, sélectionnez la flèche à côté des groupes de sécurité EC2 (pare-feu) pour développer cette section. Sous Nœud primaire et Nœuds principal et nœud de tâche, les groupes de sécurité gérés par Amazon EMR par défaut sont sélectionnés par défaut. Si vous utilisez un sous-réseau privé, vous avez également la possibilité de sélectionner un groupe de sécurité pour l'accès au service.

4. Pour modifier votre groupe de sécurité géré par Amazon EMR, utilisez le menu déroulant Choisir les groupes de sécurité pour sélectionner une autre option dans la liste des options des Groupes de sécurité gérés par Amazon EMR. Vous disposez d'un groupe de sécurité géré par Amazon EMR pour Nœud primaire et Nœuds principaux et nœuds de tâches.
5. Pour ajouter des groupes de sécurité personnalisés, utilisez le même menu déroulant Choisir les groupes de sécurité pour sélectionner jusqu'à quatre groupes de sécurité personnalisés dans la liste des options des Groupes de sécurité personnalisés. Vous pouvez avoir jusqu'à quatre groupes de sécurité personnalisés pour Nœud primaire, Nœuds principaux et nœuds de tâches.
6. Choisissez toutes les autres options qui s'appliquent à votre cluster.
7. Pour lancer votre cluster, choisissez Créer le cluster.

Old console

Pour spécifier des groupes de sécurité avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster et Go to advanced options (Aller aux options avancées).
3. Choisissez des options pour votre cluster jusqu'à ce que vous atteignez l'Step 4: Security (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 Create Cluster (Créer un cluster).

Spécifier les groupes de sécurité à l'aide de l' AWS CLI

Pour spécifier des groupes de sécurité à l'aide de la `create-cluster` commande, AWS CLI vous devez utiliser la commande avec les paramètres suivants de l'`--ec2-attributes`option :

Paramètre	Description
`EmrManagedPrimarySecurityGroup`	Utilisez ce paramètre pour spécifier un groupe de sécurité géré personnalisé pour l'instanc e principale. Si ce paramètre est spécifié, `EmrManagedCoreSecurityGroup` doit également être spécifié. Pour les clusters dans des sous-réseaux privés, `ServiceAc cessSecurityGroup` doit également être spécifié.
`EmrManagedCoreSecurityGroup`	Utilisez ce paramètre pour spécifier un groupe de sécurité géré personnalisé pour les instances principales et de tâches. Si ce paramètre est spécifié, `EmrManage dPrimarySecurityGroup` doit également être spécifié. Pour les clusters dans des sous-réseaux privés, `ServiceAc cessSecurityGroup` 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

Paramètre	Description
	groupe de sécurité que vous spécifiez comme `ServiceAccessSecurityGroup` ne doit pas être utilisé à d'autres fins et doit également être réservé à Amazon EMR. Si ce paramètre est spécifié, `EmrManagedPrimarySecurityGroup` doit également être spécifié.
`AdditionalPrimarySecurityGroups`	Utilisez ce paramètre pour spécifier jusqu'à quatre groupes de sécurité gérés personnalisés pour l'instance principale.
`AdditionalCoreSecurityGroups`	Utilisez ce paramètre pour spécifier jusqu'à quatre groupes de sécurité gérés personnalisés pour les instances principales et de tâches.

Example – spécifier 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 continuation de ligne Linux (\) sont inclus pour des raisons de lisibilité. Ils peuvent être supprimés ou utilisés dans les commandes Linux. Pour Windows, supprimez-les ou remplacez-les par un caret (^).

```
aws emr create-cluster --name "ClusterCustomManagedAndAdditionalSGs" \
--release-label emr-emr-7.1.0 --applications Name=Hue Name=Hive \
Name=Pig --use-default-roles --ec2-attributes \
SubnetIds=subnet-xxxxxxxxxxxx,KeyName=myKey,\
ServiceAccessSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedPrimarySecurityGroup=sg-xxxxxxxxxxxx,\
```

```
EmrManagedCoreSecurityGroup=sg-xxxxxxxxxxx,\
AdditionalPrimarySecurityGroups=['sg-xxxxxxxxxxx',\
'sg-xxxxxxxxxxx','sg-xxxxxxxxxxx'],\
AdditionalCoreSecurityGroups=sg-xxxxxxxxxxx \
--instance-type m5.xlarge
```

Pour plus d'informations, consultez create-cluster dans la Référence des commandes de la AWS CLI .

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 le bloc-notes EMR et le cluster Amazon EMR lorsque vous utilisez l'éditeur de bloc-notes. Les groupes de sécurité par défaut disposent de règles minimales qui autorisent uniquement le trafic réseau entre le service de 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 via le port TCP 18888. Lorsque vous créez des groupes de sécurité personnalisés avec des règles que vous adaptez à votre environnement, vous pouvez limiter le trafic réseau afin que seul un sous-ensemble de blocs-notes puisse exécuter du code dans l'éditeur de blocs-notes sur des clusters particuliers. Le cluster utilise votre sécurité personnalisée en plus des groupes de sécurité par défaut du cluster. Pour plus d'informations, consultez Contrôle du trafic réseau avec les groupes de sécurité dans le Guide de gestion Amazon EMR et Spécification des groupes de sécurité EC2 pour les blocs-notes EMR.

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 : ElasticMapReduceEditors-Livy

Règles

- 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

Aucun

Groupe de sécurité EC2 par défaut pour les blocs-notes EMR

Le groupe de sécurité EC2 par défaut pour le bloc-notes EMR est associé à l'éditeur de bloc-notes pour tout bloc-notes EMR auquel il est affecté.

Nom du groupe : ElasticMapReduceEditors-Editor

Règles

- Entrant

  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 d'acheminement vers l'extérieur afin que le bloc-notes puisse 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 appliquer les mêmes règles de sortie aux autres blocs-notes attachés à ce groupe de sécurité.

Règles

- Entrant

  Aucun
- Sortant

  Autorisez le bloc-notes à acheminer le trafic vers Internet via le cluster, comme le montre l'exemple suivant. La valeur 0.0.0.0/0 est utilisée à des fins d'exemple. Vous pouvez modifier cette règle pour spécifier les adresses IP de vos référentiels basés sur Git.

Type	Protocole	Plage de ports	Destination
Règle TCP personnalisée	TCP	18888	SG-
HTTPS	TCP	443	0.0.0.0/0

Utilisation du blocage de l'accès public Amazon EMR

Le blocage de l'accès public (BPA) d'Amazon EMR vous empêche de lancer un cluster dans un sous-réseau public si le cluster possède une configuration de sécurité qui autorise le trafic entrant depuis des adresses IP publiques sur un port.

> **Important**
>
> Le blocage de l'accès public est activé par défaut. Pour améliorer la protection du compte, nous vous recommandons de le laisser activé.

Comprendre le blocage de l'accès public

Vous pouvez utiliser la configuration au niveau du compte du blocage d'accès public pour gérer de manière centralisée l'accès au réseau public aux clusters Amazon EMR.

Lorsqu'un utilisateur de votre groupe Compte AWS lance un cluster, Amazon EMR vérifie les règles de port du groupe de sécurité du cluster et les compare à vos règles de trafic entrant. Si le groupe de sécurité dispose d'une règle entrante qui ouvre des ports aux adresses IP publiques IPv4 0.0.0.0/0 ou IPv6 ::/0, et que ces ports ne sont pas spécifiés comme des exceptions pour votre compte, Amazon EMR n'autorise pas l'utilisateur à créer le cluster.

Si un utilisateur modifie les règles du groupe de sécurité pour un cluster en cours d'exécution dans un sous-réseau public afin d'établir une règle d'accès public qui enfreint la configuration BPA de votre compte, Amazon EMR révoque la nouvelle règle s'il est autorisé à le faire. Si Amazon EMR n'est pas autorisé à révoquer la règle, il crée un événement dans le tableau de bord AWS Health qui décrit la violation. Pour accorder l'autorisation de révoquer la règle à Amazon EMR, consultez Configurer Amazon EMR pour révoquer les règles du groupe de sécurité.

Bloquer l'accès public est activé par défaut pour tous les clusters dans chaque Région AWS pour votre Compte AWS. Le BPA s'applique à l'ensemble du cycle de vie d'un cluster, mais pas aux clusters que vous créez dans des sous-réseaux privés. Vous pouvez configurer des exceptions à la règle BPA ; le port 22 est une exception par défaut. Pour plus d'informations sur la définition des exceptions, consultez Configurer le blocage de l'accès public.

Configurer le blocage de l'accès public

Vous pouvez mettre à jour les groupes de sécurité et la configuration de blocage de l'accès public dans vos comptes à tout moment.

Vous pouvez activer et désactiver les paramètres de blocage de l'accès public (BPA) à l'aide de l'API AWS Management Console, de la AWS Command Line Interface (AWS CLI) et d'Amazon EMR. Les paramètres s'appliquent à votre compte région par région. Pour garantir la sécurité du cluster, nous vous recommandons d'utiliser le BPA.

New console

> **Note**
>
> Nous avons repensé la console Amazon EMR pour la rendre plus facile à utiliser. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

Pour configurer le blocage de l'accès public avec la nouvelle console

1. Connectez-vous à la console Amazon EMR AWS Management Console, puis ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Dans la barre de navigation supérieure, sélectionnez la région que vous souhaitez configurer si elle n'est pas déjà sélectionnée.
3. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Bloquer l'accès public.
4. Sous Block public access settings (Bloquer les paramètres de blocage d'accès public), suivez les étapes ci-dessous.

Pour…	Faites ceci...

Pour…	Faites ceci...
Activer ou désactiver le blocage d'accès public	Choisissez Modifier, puis sélectionnez Activer ou Désactiver selon le cas, puis sélectionnez Enregistrer.
Modifier les ports dans la liste des exceptions	1. Choisissez Modifier et recherchez la section Exceptions relatives à la plage de ports. 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, sélectionnez Supprimer en regard de l'entrée dans la liste des plages de ports. 4. Choisissez Enregistrer.

Old console

Pour afficher, configurer, bloquer l'accès public avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/emr.
2. Dans la barre de navigation supérieure, vérifiez que la région que vous voulez 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), choisisse z 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 Save Changes (Enregistrer les modifications).

AWS CLI

Pour configurer le blocage de l'accès public à l'aide du 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. `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`

Pour…	Faites ceci...
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 } ] }'`

Configurer Amazon EMR pour révoquer les règles du groupe de sécurité

Amazon EMR a besoin d'une autorisation pour révoquer les règles du groupe de sécurité et se conformer à votre configuration de blocage de l'accès public. Vous pouvez utiliser l'une des approches suivantes pour accorder à Amazon EMR l'autorisation dont il a besoin :

- (Recommandé) Attachez la politique gérée `AmazonEMRServicePolicy_v2` au rôle de service. Pour plus d'informations, consultez Rôle de service pour Amazon EMR (rôle EMR).
- Créez une nouvelle politique intégrée qui autorise l'action `ec2:RevokeSecurityGroupIngress` sur les groupes de sécurité. Pour plus d'informations sur la façon de modifier une politique d'autorisations de rôle, consultez la section Modification d'une politique d'autorisations de rôle avec la console IAM, l'API AWS et AWS CLI dans le Guide de l'utilisateur IAM.

Résoudre les violations de blocage de l'accès public

En cas de violation du blocage de l'accès public, vous pouvez y remédier en procédant de l'une des manières suivantes :

- Si vous souhaitez accéder à une interface Web sur votre cluster, utilisez l'une des options décrites dans Affichage des interfaces Web hébergées sur des clusters Amazon EMR pour accéder à l'interface via SSH (port 22).
- Pour autoriser le trafic vers le cluster à partir d'adresses IP spécifiques plutôt que de l'adresse IP publique, ajoutez une règle de groupe de sécurité. Pour plus d'informations, consultez la section Ajout de règles à un groupe de sécurité dans le Guide de démarrage Amazon EC2.
- (Non recommandé) Vous pouvez configurer les exceptions BPA Amazon EMR pour inclure le port ou la plage de ports souhaités. Lorsque vous spécifiez une exception BPA, vous introduisez un risque avec un port non protégé. Si vous envisagez de spécifier une exception, vous devez la supprimer dès qu'elle n'est plus nécessaire. Pour plus d’informations, consultez Configurer le blocage de l'accès public.

Identifier les clusters associés aux règles des groupes de sécurité

Vous devrez peut-être identifier tous les clusters associés à une règle de groupe de sécurité donnée, ou rechercher la règle de groupe de sécurité pour un cluster donné.

- Si vous connaissez le groupe de sécurité, vous pouvez identifier les clusters associés si vous trouvez les interfaces réseau du groupe de sécurité. Pour plus d'informations, consultez Comment puis-je trouver les ressources associées à un groupe de sécurité Amazon EC2 ? sur AWS re:Post. Les instances Amazon EC2 associées à ces interfaces réseau seront étiquetées avec l'ID du cluster auquel elles appartiennent.
- Si vous souhaitez rechercher les groupes de sécurité d'un cluster connu, suivez les étapes décrites dans Afficher l'état et les détails d'un cluster. Vous trouverez les groupes de sécurité du cluster dans le panneau Réseau et sécurité de la console, ou dans le champ `Ec2InstanceAttributes` de l' AWS CLI.

Validation de la conformité d'Amazon EMR

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon EMR dans le cadre de plusieurs programmes de AWS conformité. Il s’agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d’autres.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformité. Pour obtenir des informations générales, veuillez consulter Programmes de conformité d'AWS.

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, consultez la section Téléchargement de 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 d'Amazon EMR est soumise au respect de normes telles que HIPAA, PCI ou FedRAMP, fournit des ressources pour vous aider à : AWS

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- Livre blanc Architecting for HIPAA Security and Compliance — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS ressources relatives à la conformité : cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Config— Ce AWS service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans Amazon EMR

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone 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 AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Outre l'infrastructure AWS mondiale, Amazon EMR propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des 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 la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon EMR via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

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

Rubriques

- Connexion à Amazon EMR à l'aide d'un point de terminaison d'un VPC d'interface

Connexion à Amazon EMR à l'aide d'un point de terminaison d'un VPC d'interface

Vous pouvez vous connecter directement à Amazon EMR à l'aide d'un point de terminaison 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 VPC d'interface, la communication entre votre VPC et Amazon EMR s'effectue 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 VPC de l'interface connecte votre VPC directement à Amazon EMR sans passerelle Internet, périphérique NAT, connexion VPN ou 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 réseau privé virtuel (VPN) Amazon ou du AWS Direct Connect. Pour obtenir des informations sur Amazon VPN, consultez la rubrique Connexions VPN du Guide de l'utilisateur Amazon Virtual Private Cloud. Pour plus d'informations AWS Direct Connect, voir Création d'une connexion dans le Guide de AWS Direct Connect l'utilisateur.

Vous pouvez créer un point de terminaison VPC d'interface pour vous connecter à Amazon EMR à l'aide de la AWS console ou AWS Command Line Interface des commandes ().AWS CLI Pour plus d'informations, consultez Création d'un point de terminaison d'interface.

Une fois que vous avez créé un point de terminaison d'un 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 plus d'informations, consultez la section Interface VPC endpoints (AWS PrivateLink) dans le guide de l'utilisateur Amazon VPC.

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 politiques de point de terminaison d'un VPC au point de terminaison d'un VPC pour contrôler l'accès des principaux 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, veuillez consulter Contrôler l'accès aux services avec les points de terminaison d'un VPC.

Créez une politique de point de terminaison d'un VPC pour Amazon EMR.

Vous pouvez créer une politique pour les points de terminaison de VPC Amazon 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 plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Example — Politique de point de terminaison VPC interdisant tout accès depuis un compte spécifié AWS

La politique de point de terminaison VPC suivante refuse au AWS compte *123456789012* tout accès aux ressources utilisant le point de terminaison.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example – Politique du point de terminaison d'un VPC pour autoriser l'accès VPC uniquement à un principal (utilisateur) IAM spécifié

La politique de point de terminaison VPC suivante autorise un accès complet uniquement à un utilisateur lijuan dans AWS le compte 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 d'un VPC pour autoriser les opérations EMR en lecture seule

La politique de point de terminaison VPC suivante autorise uniquement le AWS compte *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 obtenir la liste des actions Amazon EMR, consultez 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 d'un VPC refusant l'accès à un cluster spécifié

La politique de point de terminaison VPC suivante autorise un accès complet à tous les comptes et à tous les principaux, mais refuse tout accès pour le AWS compte 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"
                ]
            }
        }
    ]
}
```

Gestion des clusters

Après avoir 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

- Connexion à un cluster
- Soumission de travail à un cluster
- Affichage et surveillance d'un cluster
- Utiliser la mise à l'échelle des clusters
- Arrêter un cluster
- Clonage d'un cluster à l'aide de la console
- Automatisation de clusters récurrents avec AWS Data Pipeline

Connexion à un cluster

Lorsque vous exécutez un cluster Amazon EMR, il vous suffit souvent d'exécuter une application pour analyser vos données, puis de collecter les données de sortie à partir d'un compartiment Amazon S3. A d'autres moments, vous pouvez souhaiter interagir avec le nœud primaire alors que le cluster est en cours d'exécution. Par exemple, vous pouvez souhaiter vous connecter au nœud primaire 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 primaire, etc. Les sections suivantes décrivent les techniques que vous pouvez utiliser pour vous connecter au nœud primaire.

Dans un cluster EMR, le nœud primaire est une instance Amazon EC2 qui coordonne les instances EC2 s'exécutant en tant que nœuds de tâches et nœuds principaux. Le nœud primaire 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 primaire, les nœuds principaux et les tâches ; elles déterminent la façon dont vous accédez aux nœuds.

> **Note**
>
> Vous pouvez vous connecter au nœud primaire uniquement pendant l'exécution d'un cluster. Lorsque le cluster prend fin, l'instance EC2 qui agit en tant que nœud primaire est mise

hors service et n'est plus disponible. Pour vous connecter au nœud primaire, 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 Utilisation de Kerberos pour l'authentification avec Amazon EMR. 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 des groupes de sécurité, consultez la section Ajouter des règles à un groupe de sécurité dans le guide de l'utilisateur Amazon EC2.

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

- Avant de vous connecter : autoriser le trafic entrant
- Connexion au nœud primaire à l'aide de SSH

Avant de vous connecter : autoriser le trafic entrant

Avant de vous connecter à un cluster Amazon EMR, vous devez autoriser le trafic SSH entrant (port 22) en provenance de clients fiables tels que l'adresse IP de votre ordinateur. Pour ce faire, modifiez les règles des groupes de sécurité gérés pour les nœuds auxquels vous souhaitez vous connecter. Par exemple, les instructions suivantes vous montrent comment ajouter une règle entrante pour l'accès SSH au groupe de sécurité ElasticMapReduce -master par défaut.

Pour de plus amples informations sur les groupes de sécurité avec Amazon EMR, consultez Contrôle du trafic réseau avec des groupes de sécurité.

New console

Accorder à des sources fiables l'accès SSH au groupe de sécurité principal à l'aide de la nouvelle console

Pour modifier vos groupes de sécurité, vous devez avoir l'autorisation de gérer les groupes de sécurité pour le VPC dans lequel se trouve le cluster. Pour plus d'informations, consultez Modification des autorisations d'un utilisateur et l'exemple de politique permettant de gérer les groupes de sécurité EC2 dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Dans le volet de navigation de gauche, sous EMR on EC2, choisissez Clusters, puis le cluster que vous souhaitez mettre à jour. La page de détails du cluster s'ouvre. L'onglet Propriétés de cette page est présélectionné.
3. Sous Mise en réseau dans l'onglet Propriétés, sélectionnez la flèche à côté des groupes de sécurité EC2 (pare-feu) pour développer cette section. Sous Nœud primaire, sélectionnez le lien du groupe de sécurité. Ceci ouvre la console EC2.
4. Sélectionnez l'onglet Règles entrantes, puis Modifier les règles entrantes.
5. Vérifiez s'il existe une règle entrante qui autorise l'accès public avec les paramètres suivants. Si elle existe, choisissez Supprimer pour la supprimer.

   - Type

     SSH
   - Port

     22
   - Source

     Personnalisé 0.0.0.0/0

   > **Warning**
   >
   > Avant décembre 2020, le groupe de sécurité ElasticMapReduce -master disposait d'une règle préconfigurée pour autoriser le trafic entrant sur le port 22 en provenance de toutes les sources. Cette règle a été créée pour simplifier les connexions SSH

initiales au nœud primaire. Nous vous recommandons vivement de supprimer cette règle d'entrée et de limiter le trafic aux sources fiables.

6. Faites défiler la liste des règles jusqu'en bas et sélectionnez Ajouter une règle.
7. Dans le champ Type, sélectionnez SSH. Cette sélection saisit automatiquement TCP pour le protocole et 22 pour la plage de ports.
8. Pour source, sélectionnez Mon adresse IP pour ajouter automatiquement votre adresse IP en tant qu'adresse source. Vous pouvez également ajouter une plage d'adresses IP de clients fiables personnalisées ou créer des règles supplémentaires pour d'autres clients. De nombreux environnements réseau allouent des adresses IP de manière dynamique. Il se peut donc que vous deviez mettre à jour vos adresses IP pour les clients fiables à l'avenir.
9. Choisissez Enregistrer.
10. Retournez éventuellement à l'étape 3, choisissez les nœuds de tâches et principaux, puis répétez les étapes 4 à 8. Cela permet aux nœuds principaux et aux nœuds de tâches d'accéder au client SSH.

Old console

Pour accorder à des sources fiables un accès SSH au groupe de sécurité principal via la console

Pour modifier vos groupes de sécurité, vous devez avoir l'autorisation de gérer les groupes de sécurité pour le VPC dans lequel se trouve le cluster. Pour plus d'informations, consultez Modification des autorisations d'un utilisateur et l'exemple de politique permettant de gérer les groupes de sécurité EC2 dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Clusters. Choisissez l'ID du cluster que vous souhaitez modifier.
3. Dans le volet Réseau et sécurité, développez la liste déroulante des groupes de sécurité EC2 (pare-feu).
4. Sous Nœud principal, choisissez votre groupe de sécurité.
5. Choisissez Modifier les règles entrantes.
6. Vérifiez s'il existe une règle entrante qui autorise l'accès public avec les paramètres suivants. Si elle existe, choisissez Supprimer pour la supprimer.

   - Type

SSH

- Port

  22

- Source

  Personnalisé 0.0.0.0/0

> **Warning**
>
> Avant décembre 2020, une règle préconfigurée autorisait le trafic entrant sur le port 22 en provenance de toutes les sources. Cette règle a été créée pour simplifier les connexions SSH initiales au nœud primaire. Nous vous recommandons vivement de supprimer cette règle d'entrée et de limiter le trafic aux sources fiables.

7. Faites défiler la liste des règles jusqu'en bas et sélectionnez Ajouter une règle.
8. Dans le champ Type, sélectionnez SSH.

   En sélectionnant SSH, vous saisissez automatiquement TCP pour le protocole et 22 pour la plage de ports.
9. Pour source, sélectionnez Mon adresse IP pour ajouter automatiquement votre adresse IP en tant qu'adresse source. Vous pouvez également ajouter une plage d'adresses IP de clients fiables personnalisées ou créer des règles supplémentaires pour d'autres clients. De nombreux environnements réseau allouent des adresses IP de manière dynamique. Il se peut donc que vous deviez mettre à jour vos adresses IP pour les clients fiables à l'avenir.
10. Choisissez Enregistrer.
11. Choisissez éventuellement l'autre groupe de sécurité sous Nœuds principaux et de tâches dans le volet Réseau et sécurité et répétez les étapes ci-dessus pour autoriser l'accès du client SSH aux nœuds principaux et aux nœuds de tâches.

Connexion au nœud primaire 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, qui est un serveur virtuel exécuté 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 primaire du cluster.

Lorsque vous utilisez SSH pour vous connecter au nœud primaire, vous pouvez surveiller le cluster et interagir avec lui. Vous pouvez émettre des commandes Linux sur le nœud primaire, 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 primaire. Pour plus d'informations, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Pour vous connecter au nœud primaire à l'aide de SSH, vous avez besoin du nom DNS public du nœud primaire. En outre, le groupe de sécurité rattaché au nœud primaire 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 plus d'informations sur la modification des règles des groupes de sécurité, consultez Contrôle du trafic réseau avec des groupes de sécurité la section Ajouter des règles à un groupe de sécurité dans le guide de l'utilisateur Amazon EC2.

Récupération du nom DNS public du nœud primaire

Vous pouvez récupérer le nom de serveur DNS public du nœud primaire à l'aide de la console Amazon EMR et de l'interface AWS CLI.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Récupérer le nom DNS public du nœud primaire à l'aide de la nouvelle console

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

2. Sous EMR on EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster dans lequel vous souhaitez récupérer le nom DNS public.
3. Notez la valeur DNS public du nœud primaire dans la section Récapitulatif de la page de détails du cluster.

Old console

Récupérer le nom DNS public du nœud primaire à l'aide de l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Dans la page Liste de clusters, sélectionnez le lien de votre cluster.
3. Notez la valeur DNS public du nœud primaire qui s'affiche dans la partie Récapitulatif de la page Détails du cluster.

> **Note**
>
> Vous pouvez également choisir le lien SSH pour obtenir des informations sur la création d'une connexion SSH avec le nœud primaire.

CLI

Pour récupérer le nom DNS public du nœud principal à l'aide du 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 cluster, tapez l'une des commandes suivantes. Remplacez *j-2AL4XXXXXX5T9* par 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 pour `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 sur l'utilisation des commandes, consultez Commandes Amazon EMR dans l'interface AWS CLI.

Connexion au nœud primaire 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. Pour plus d'informations sur l'accès à votre paire de clés, consultez les paires de clés Amazon EC2 dans le guide de l'utilisateur Amazon EC2.

Votre ordinateur Linux inclut très probablement 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 primaire. Le projet OpenSSH offre une implémentation gratuite de la suite entière des outils SSH. Pour plus d'informations, consultez le site Web OpenSSH.

Les instructions suivantes décrivent l'ouverture d'une connexion SSH sur le nœud primaire 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. Cela est nécessaire pour créer une connexion SSH à l'aide du terminal ou du AWS CLI.

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. 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.
3. Pour définir les autorisations, saisissez la commande suivante. Remplacez *~/mykeypair.pem* par le chemin d'accès complet et le nom de votre fichier de clé privée rattaché à votre paire de clés. Par exemple `C:/Users/<username>/.ssh/mykeypair.pem`.

   ```
   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.

Connexion au nœud primaire 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 primaire, tapez la commande suivante. Remplacez *ec2-###-####.compute-1.amazonaws.com* par le nom DNS public du nœud primaire de votre cluster et remplacez *~/mykeypair.pem* par le chemin d'accès complet et le nom de votre fichier `.pem`. Par exemple `C:/Users/<username>/.ssh/mykeypair.pem`.

   ```
   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 primaire 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 primaire, tapez la commande suivante pour fermer la connexion SSH.

   ```
   exit
   ```

Si vous rencontrez des difficultés pour utiliser SSH lors de la connexion à votre nœud primaire, consultez Résoudre les problèmes de connexion à votre instance.

Connexion au nœud primaire à l'aide de SSH sous Windows

Les utilisateurs Windows peuvent utiliser un client SSH tel que PuTTY pour se connecter au nœud primaire. Avant de vous connecter au nœud primaire Amazon EMR, vous devez télécharger et

installer PuTTY ainsi que 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`). Avant d'essayer de vous connecter au nœud primaire en utilisant PuTTY, vous devez convertir votre clé dans ce format (`.ppk`).

Pour plus d'informations sur la conversion de votre clé, consultez la section Conversion de votre clé privée à l'aide de PuttyGen dans le guide de l'utilisateur Amazon EC2.

Connexion au nœud primaire à l'aide de PuTTY

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. Ouvrir `putty.exe`. Vous pouvez également lancer PuTTY à partir de la liste des programmes Windows.
3. Si nécessaire, dans la liste Category (Catégorie), choisissez Session.
4. Dans Nom d'hôte (ou adresse IP), tapez `hadoop@` *`MasterPublicDNS`*. Par exemple : `hadoop@`*`ec2-###-##-##-###.compute-1.amazonaws.com`*.
5. Dans la liste Category (Catégorie), sélectionnez Connection > SSH (Connexion > SSH), Auth.
6. 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.
7. Choisissez Ouvrir et Oui pour ignorer l'alerte de sécurité PuTTY.

> **Important**
>
> Lorsque vous vous connectez au nœud primaire, tapez `hadoop` si vous êtes invité à saisir un nom d'utilisateur.

8. Lorsque vous avez terminé d'utiliser le nœud primaire, 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 menu contextuel (clic droit) pour la barre de titre PuTTY et en choisissant Modifier les paramètres.

Si vous rencontrez des difficultés pour utiliser SSH lors de la connexion à votre nœud primaire, consultez Résoudre les problèmes de connexion à votre instance.

Connexion au nœud primaire à l'aide de l'interface AWS CLI

Vous pouvez créer une connexion SSH avec le nœud principal à l' AWS CLI aide de Windows et Linux, Unix et Mac OS X. Quelle que soit la plateforme, vous avez besoin du nom DNS public du nœud principal et de votre clé privée de paire de clés Amazon EC2. Si vous utilisez AWS CLI le sous Linux, Unix ou Mac OS X, vous devez également définir des autorisations sur le fichier de clé privée (`.pemou.ppk`) comme indiqué dansPour configurer les autorisations sur les fichiers de clé privée de paire de clés.

Pour vous connecter au nœud principal à l'aide du AWS CLI

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
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 une connexion SSH vers le nœud primaire. Dans l'exemple suivant, remplacez *j-2AL4XXXXXX5T9* par l'ID du cluster et remplacez *~/mykeypair.key* par le chemin d'accès complet et le nom de votre fichier `.pem` (pour Linux, Unix et Mac OS X) ou `.ppk` (pour Windows). Par exemple `C:\Users\<username>\.ssh\mykeypair.pem`.

```
aws emr ssh --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

4. Lorsque vous avez terminé de travailler sur le nœud principal, fermez la AWS CLI fenêtre.

Pour plus d'informations sur l'utilisation des commandes, consultez Commandes Amazon EMR dans l'interface AWS CLI. Si vous rencontrez des difficultés pour utiliser SSH lors de la connexion à votre nœud primaire, consultez Résoudre les problèmes de connexion à votre instance.

Ports de service Amazon EMR

> **Note**
>
> Vous trouverez ci-dessous les interfaces et les ports de service pour les composants sur Amazon EMR. Il ne s'agit pas d'une liste complète des ports de service. Les services autres que ceux par défaut, tels que les ports SSL et les différents types de protocoles, ne sont pas répertoriés.

> **Important**
>
> Soyez vigilant lorsque vous modifiez les règles du groupe de sécurité pour ouvrir des ports. Assurez-vous d'ajouter des règles qui autorisent uniquement le trafic provenant de clients approuvés et authentifiés pour les protocoles et les ports nécessaires à l'exécution de vos charges de travail.

Composant	Service description (Description du service)	Service exécuté par défaut	Port	Clé de configuration
Hadoop	HTTP KMS API REST	Oui	9600	hadoop.kms.http.port
HDFS	Interface utilisateur Web NameNode	Oui	9870	dfs.namenode.http-address
	NameNode RPC	Oui	8020	dfs.namenode.rpc-address
	DataNode Interface utilisateur Web	Oui	9864	dfs.datanode.http.address
	Datanode HTTP pour le transfert de données	Oui	9866	dfs.datanode.address
	Datanode RPC pour le transfert de données	Oui	9867	dfs.datanode.ipc.address
Hive	HiveServer2 Épargne	Oui	10 000	hive.server2.thrift.port
	HiveServer2 HTTP	Non	10001	hive.server2.thrift.http.port
	HiveServer2 Interface utilisateur Web	Oui	10002	hive.server2.webui.port
	Hive Metastore	Oui	9083	hive.metastore.port / metastore.thrift.port

Composant	Service description (Description du service)	Service exécuté par défaut	Port	Clé de configuration
	WebHcat	Non	50111	templeton.port
	Service de gestion des démons LLAP (RPC)	Non	15004	hive.llap.management.rpc.port
	Port YARN shuffle pour le shuffle hébergé par LLAP démon	Non	15551	hive.llap.daemon.yarn.shuffle.port
	Le RPC démon LLAP	Non	Répartition dynamique	hive.llap.daemon.rpc.port
	Interface utilisateur Web du démon LLAP	Non	15002	hive.llap.daemon.web.port
	Service de sortie du démon LLAP	Non	15003	hive.llap.daemon.output.service.port
Oozie		Oui	11 000	
Tez	Interface utilisateur Tez	Oui	8080	
YARN	Shuffle	Oui	13562	mapreduce.shuffle.port
	Localisateur RPC	Oui	8040	yarn.nodemanager.localizer.address
		Oui	8041	

Composant	Service description (Description du service)	Service exécuté par défaut	Port	Clé de configuration
	Adresse de l'application Web NM	Oui	8042	yarn.nodemanager.webapp.address
	Application Web RM	Oui	8088	yarn.resourcemanager.webapp.address
		Oui	8025	
	Planificateur	Oui	8030	yarn.resourcemanager.scheduler.address
	interface du gestionnaire d'applications	Oui	8032	yarn.resourcemanager.address
	Interface d'administration RM	Oui	8033	yarn.resourcemanager.admin.address
	JobHistory Interface utilisateur Web du serveur	Oui	1988	mapreduce.jobhistory.webapp.address
	JobHistory Interface utilisateur Web d'administration du serveur	Oui	10033	mapreduce.jobhistory.admin.address

Composant	Service description (Description du service)	Service exécuté par défaut	Port	Clé de configuration
	JobHistory Serveur (RPC)	Oui	10020	mapreduce.jobhistory.address
	Serveur de chronologie des applications (RPC)	Oui	10200	yarn.timeline-service.address
	Interface utilisateur Web HTTP du serveur de chronologie des applications	Oui	8188	yarn.timeline-service.webapp.address
	Interface utilisateur Web HTTPS du serveur de chronologie des applications	Non	8190	yarn.timeline-service.webapp.https.address
		Oui	20888	
Zookeeper	Port client	Oui	2181	
		Oui	37301	
		Oui	8341	

Affichage des interfaces Web hébergées sur des clusters Amazon EMR

> **Important**
>
> 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 Contrôle du trafic réseau avec des groupes de sécurité.

Hadoop et les autres applications que vous installez sur votre cluster EMR publient des interfaces utilisateur en tant que sites Web hébergés sur le nœud primaire. Pour des raisons de sécurité, lors de l'utilisation des groupes de sécurité gérés par Amazon EMR, ces sites Web sont uniquement disponibles sur le serveur Web local du nœud primaire et, par conséquent, vous devez vous connecter au nœud primaire pour les afficher. Ainsi, vous devez vous connecter au nœud primaire pour afficher les interfaces Web. Pour plus d'informations, consultez Connexion au nœud primaire à l'aide de SSH. 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.

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'instances principales, remplacez *master-public-dns-name* par le DNS public principal répertorié sur l'onglet de cluster Récapitulatif dans la console Amazon EMR. Pour les interfaces d'instances principales et de tâches, remplacez *coretask-public-dns-name* par le Public DNS name (Nom du DNS public) répertorié pour l'instance. Pour rechercher un Nom de DNS public, dans la console Amazon EMR, sélectionnez votre cluster dans la liste, choisissez l'onglet Matériel, choisissez l'ID du groupe d'instances qui contient l'instance à laquelle vous souhaitez vous connecter, puis notez le Nom de DNS public répertorié pour l'instance.

Nom de l'interface	URI
Serveur d'historique Flink (EMR version 5.33 et versions ultérieures)	http://*master-public-dns-name* :8082/
Ganglia	http://*master-public-dns-name* /ganglia/
Hadoop HDFS (version NameNode EMR antérieure à 6.x)	https://*master-public-dns-name* :50470/
Hadoop HDFS NameNode	http://*master-public-dns-name* :50070/

Nom de l'interface	URI
Hadoop HDFS DataNode	http://*coretask-public-dns-name* :50075/
Hadoop HDFS (NameNode EMR version 6.x)	https://*master-public-dns-name* :9870/
Hadoop HDFS (version DataNode EMR antérieure à 6.x)	https://*coretask-public-dns-name* :50475/
Hadoop HDFS (DataNode 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/
Étincelle HistoryServer	http://*master-public-dns-name* :18080/
Tez	http://*master-public-dns-name* :8080/tez-ui
LAINE NodeManager	http://*coretask-public-dns-name* :8042/
LAINE ResourceManager	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 primaire, mais ne sont pas disponibles sur les nœuds principaux et de tâches, les instructions de ce document sont spécifiques au nœud primaire 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 primaire.

Il existe plusieurs façons d'accéder aux interfaces Web sur le nœud primaire. La méthode la plus simple et la plus rapide consiste à utiliser SSH pour vous connecter au nœud primaire 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' ResourceManager interface Hadoop à l'aide de Lynx (les URL Lynx sont également fournies lorsque vous vous connectez au nœud principal 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 primaire, qui fournissent des fonctionnalités de navigateur complet. Sélectionnez l'une des méthodes suivantes :

- Option 1 (recommandée pour les utilisateurs plus techniques) : utilisez un client SSH pour vous connecter au nœud primaire, configurez le tunnel SSH avec le réacheminement de port local et utilisez un navigateur Internet pour ouvrir les interfaces Web hébergées sur le nœud primaire. 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 principal, configurez le tunneling SSH avec redirection de port dynamique et configurez votre navigateur Internet pour utiliser un module complémentaire tel que Firefox ou Chrome FoxyProxy SwitchyOmega pour gérer les paramètres de votre 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 primaire. Pour plus d'informations sur la configuration FoxyProxy de Firefox et Google Chrome, consultezOption 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire.

> **Note**
>
> Si vous modifiez le port sur lequel une application s'exécute via la configuration du cluster, le lien hypertexte vers le port ne sera pas mis à jour dans la console Amazon EMR. Cela est dû au fait que la console ne dispose pas de la fonctionnalité permettant de lire la configuration `server.port`.

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 primaire à l'aide du réacheminement de port local
- Option 2, partie 1 : Configuration d'un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port dynamique
- Option 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire

Option 1 : Configuration d'un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port local

Pour vous connecter au serveur Web local sur le nœud primaire, vous créez un tunnel SSH entre votre ordinateur et le nœud primaire. 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 primaire 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 primaire.

La configuration d'un tunnel SSH à l'aide du réacheminement de port local nécessite le nom de serveur DNS public du nœud primaire 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 Récupérer le nom DNS public du nœud primaire à l'aide de l'ancienne console . Pour plus d'informations sur l'accès à votre paire de clés, consultez les paires de clés Amazon EC2 dans le guide de l'utilisateur Amazon EC2. Pour plus d'informations sur les sites que vous pouvez afficher sur le nœud primaire, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Configurer un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port local avec OpenSSH

Pour configurer un tunnel SSH à l'aide du réacheminement de port local dans le terminal

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. 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).

3. Tapez la commande suivante pour ouvrir un tunnel SSH sur votre machine locale. Cet exemple de commande accède à l'interface ResourceManager Web en transférant le trafic sur le port local 8157 (un port local inutilisé choisi au hasard) vers le port 8088 sur le serveur Web local du nœud maître.

   Dans la commande, remplacez *~/mykeypair.pem* par l'emplacement et le nom de fichier de votre fichier `.pem` et remplacez *ec2-###-#-#-##-##.compute-1.amazonaws.com* par le nom DNS public principal de votre cluster. Pour accéder à une autre interface Web, remplacez `8088` par le numéro de port approprié. Par exemple, remplacez `8088` par `8890` pour l'interface Zeppelin.

   ```
   ssh -i ~/mykeypair.pem -N -L 8157:ec2-###-##-##-
   ###.compute-1.amazonaws.com:8088 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   `-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 principal.

   Lorsque cette commande est émise, le terminal reste ouvert et ne retourne pas de réponse.
4. Pour ouvrir l'interface ResourceManager Web de votre navigateur, tapez `http://localhost:8157/` dans la barre d'adresse.
5. Lorsque vous avez terminé d'utiliser les interfaces Web sur le nœud primaire, fermez la fenêtre du terminal.

Option 2, partie 1 : Configuration d'un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port dynamique

Pour vous connecter au serveur Web local sur le nœud primaire, vous créez un tunnel SSH entre votre ordinateur et le nœud primaire. 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 primaire. Cette action crée un proxy SOCKS. Vous pouvez ensuite configurer votre navigateur Internet pour utiliser un module complémentaire tel que FoxyProxy ou SwitchyOmega pour gérer les paramètres de votre proxy SOCKS.

À 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 primaire. 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 primaire et ceux hébergés sur Internet.

Avant de commencer, vous devez connaître le nom de serveur DNS public du nœud primaire 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 primaire, consultez Récupérer le nom DNS public du nœud primaire à l'aide de l'ancienne console . Pour plus d'informations sur l'accès à votre paire de clés, consultez les paires de clés Amazon EC2 dans le guide de l'utilisateur Amazon EC2. Pour plus d'informations sur les sites que vous pouvez afficher sur le nœud primaire, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Configurer un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port dynamique avec OpenSSH

Configurer un tunnel SSH à l'aide du réacheminement de port dynamique avec OpenSSH

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. 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).
3. Tapez la commande suivante pour ouvrir un tunnel SSH sur votre ordinateur local. Remplacez *~/mykeypair.pem* par l'emplacement et le nom de fichier de votre `.pem` fichier, remplacez *8157* par un numéro de port local non utilisé et remplacez *ec2-####-#-#-##.compute-1.amazonaws.com* par 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 primaire. Le réacheminement de port dynamique crée un proxy SOCKS local qui écoute sur le port spécifié dans la commande.

4. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez Option 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire.
5. Lorsque vous avez terminé d'utiliser les interfaces Web sur le nœud primaire, fermez la fenêtre du terminal.

Configurez un tunnel SSH à l'aide de la redirection de port dynamique avec AWS CLI

Vous pouvez créer une connexion SSH avec le nœud principal sous Windows et sous Linux, Unix et Mac OS X. Si vous utilisez le AWS CLI nœud sous Linux, Unix ou Mac OS X, vous devez définir des autorisations sur le `.pem` fichier comme indiqué dans. AWS CLI Pour configurer les autorisations sur les fichiers de clé privée de paire de clés Si vous utilisez AWS CLI le sous Windows, PuTTY doit apparaître dans la variable d'environnement path, sinon vous risquez de recevoir un message d'erreur tel que OpenSSH ou PuTTY non disponible.

Pour configurer un tunnel SSH à l'aide de la redirection de port dynamique avec AWS CLI

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. Créez une connexion SSH avec le nœud primaire, comme illustré dans Connexion au nœud primaire à l'aide de l'interface AWS CLI.
3. 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"
```

4. Tapez la commande suivante pour ouvrir un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port dynamique. Dans l'exemple suivant, remplacez *j-2AL4XXXXXX5T9* par l'ID du cluster et remplacez *~/mykeypair.key* par l'emplacement et le nom de votre fichier `.pem` (pour Linux, Unix et Mac OS X) ou `.ppk` (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.

5. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez Option 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire.
6. Lorsque vous avez terminé d'utiliser les interfaces Web sur le nœud principal, fermez la AWS CLI fenêtre.

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

Créer un tunnel SSH vers le nœud primaire à l'aide de PuTTY

Les utilisateurs Windows peuvent utiliser un client SSH tel que PuTTY pour créer un tunnel SSH vers le nœud primaire. Avant de vous connecter au nœud primaire Amazon EMR, vous devez télécharger et installer PuTTY ainsi que 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`). Avant d'essayer de vous connecter au nœud primaire en utilisant PuTTY, vous devez convertir votre clé dans ce format (`.ppk`).

Pour plus d'informations sur la conversion de votre clé, consultez la section Conversion de votre clé privée à l'aide de PuttyGen dans le guide de l'utilisateur Amazon EC2.

Configurer un tunnel SSH à l'aide du réacheminement de port dynamique à l'aide de PuTTY

1. Assurez-vous d'avoir autorisé le trafic SSH entrant. Pour obtenir des instructions, veuillez consulter Avant de vous connecter : autoriser le trafic entrant.
2. 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 primaire, vous pouvez ajouter un tunnel en cliquant avec le bouton droit sur la barre de titre PuTTY et en choisissant Modifier les paramètres.

3. Si nécessaire, dans la liste Category (Catégorie), choisissez Session.
4. Dans le champ Nom d'hôte, tapez **hadoop@** *MasterPublicDNS*. Par exemple : **hadoop@***ec2-###-##-##-###.compute-1.amazonaws.com*.
5. Dans la liste Category (Catégorie), développez Connection > SSH (Connexion > SSH), puis choisissez Auth.
6. 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`). Avant d'essayer de vous connecter au nœud primaire en utilisant PuTTY, vous devez convertir votre clé dans ce format (`.ppk`).

7. Dans la liste Category (Catégorie), développez Connection > SSH (Connexion > SSH), puis choisissez Tunnels.
8. Dans le champ Port source, saisissez `8157` (un port local inutilisé), puis choisissez Ajouter.
9. Laissez le champ Destination vide.
10. Sélectionnez les options Dynamic (Dynamique) et Auto.

11. Choisissez Ouvrir.
12. Choisissez Yes (Oui) pour ignorer l'alerte de sécurité PuTTY.

> **Important**
>
> Lorsque vous vous connectez au nœud primaire, tapez hadoop si vous êtes invité à saisir un nom d'utilisateur.

13. Une fois que le tunnel est actif, configurez un proxy SOCKS pour votre navigateur. Pour plus d'informations, consultez Option 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire.
14. Lorsque vous avez terminé d'utiliser les interfaces Web sur le nœud primaire, fermez la fenêtre PuTTY.

Option 2, partie 2 : Configuration des paramètres de proxy pour afficher les sites Web hébergés sur le nœud primaire

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. À 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 primaire. 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 primaire et ceux hébergés sur Internet. Pour gérer vos paramètres de proxy, configurez votre navigateur pour utiliser un module complémentaire tel que FoxyProxy ou SwitchyOmega.

Pour plus d'informations sur la création d'un tunnel SSH, consultez Option 2, partie 1 : Configuration d'un tunnel SSH vers le nœud primaire à l'aide du réacheminement de port dynamique. Pour plus d'informations sur les interfaces Web disponibles, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Incluez les paramètres suivants lorsque vous configurez votre module complémentaire de proxy :

- Utilisez localhost comme adresse d'hôte.
- Utilisez le même numéro de port local que celui que vous avez sélectionné pour établir le tunnel SSH avec le nœud primaire Option 2, partie 1 : Configuration d'un tunnel SSH vers le nœud

primaire à l'aide du réacheminement de port dynamique. Par exemple, le port *8157*. 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.

- Spécifiez le protocole SOCKS v5. SOCKS v5 vous permet de configurer éventuellement l'autorisation des utilisateurs.
- Modèles URL

  Les modèles URL suivants doivent être sur liste blanche et spécifiés par un type de modèle de caractère générique :

  - Les modèles *ec2*.*compute*.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.
  - Un 10. * modèle permettant d'accéder aux fichiers JobTracker journaux dans Hadoop. Modifiez ce filtre s'il est en conflit avec votre plan d'accès réseau.
  - Les modèles *.ec2.internal* et *.compute.internal* correspondent aux noms DNS privés (internes) des clusters de la région `us-east-1` et de toutes les autres régions, respectivement.

Exemple : Configuration FoxyProxy pour Firefox

L'exemple suivant illustre une configuration FoxyProxy Standard (version 7.5.1) pour Mozilla Firefox.

FoxyProxy fournit un ensemble d'outils de gestion de proxy. Il vous permet d'utiliser un serveur proxy pour les URL qui correspondent aux modèles rattachés aux domaines utilisés par les instances Amazon EC2 de votre cluster Amazon EMR.

Pour installer et configurer à FoxyProxy l'aide de Mozilla Firefox

1. Dans Firefox, rendez-vous sur https://addons.mozilla.org/, recherchez FoxyProxy Standard et suivez les instructions pour ajouter FoxyProxy à Firefox.
2. À l'aide d'un éditeur de texte, créez un fichier JSON nommé `foxyproxy-settings.json` grâce à l'exemple de configuration suivant.

```
{
  "k20d21508277536715": {
    "active": true,
    "address": "localhost",
```

```
    "port": 8157,
    "username": "",
    "password": "",
    "type": 3,
    "proxyDNS": true,
    "title": "emr-socks-proxy",
    "color": "#0055E5",
    "index": 9007199254740991,
    "whitePatterns": [
      {
        "title": "*ec2*.*compute*.amazonaws.com*",
        "active": true,
        "pattern": "*ec2*.*compute*.amazonaws.com*",
        "importedPattern": "*ec2*.*compute*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*ec2*.compute*",
        "active": true,
        "pattern": "*ec2*.compute*",
        "importedPattern": "*ec2*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "10.*",
        "active": true,
        "pattern": "10.*",
        "importedPattern": "http://10.*",
        "type": 1,
        "protocols": 2
      },
      {
        "title": "*10*.amazonaws.com*",
        "active": true,
        "pattern": "*10*.amazonaws.com*",
        "importedPattern": "*10*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*10*.compute*",
        "active": true,
```

```
        "pattern": "*10*.compute*",
        "importedPattern": "*10*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*.compute.internal*",
        "active": true,
        "pattern": "*.compute.internal*",
        "importedPattern": "*.compute.internal*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*.ec2.internal* ",
        "active": true,
        "pattern": "*.ec2.internal*",
        "importedPattern": "*.ec2.internal*",
        "type": 1,
        "protocols": 1
      }
    ],
    "blackPatterns": []
  },
  "logging": {
    "size": 100,
    "active": false
  },
  "mode": "patterns",
  "browserVersion": "68.12.0",
  "foxyProxyVersion": "7.5.1",
  "foxyProxyEdition": "standard"
}
```

3. Ouvrez la page Gérer vos extensions de Firefox (allez sur about:addons, puis choisissez Extensions).
4. Choisissez FoxyProxy Standard, puis cliquez sur le bouton Autres options (le bouton qui ressemble à des points de suspension).
5. Sélectionnez Options dans le menu déroulant.
6. Choisissez Paramètres d'importation dans le menu de gauche.
7. Sur la page Paramètres d'importation, choisissez Paramètres d'importation sous Paramètres d'importation à partir de la FoxyProxy version 6.0, naviguez jusqu'à l'emplacement du

`foxyproxy-settings.json` fichier que vous avez créé, sélectionnez le fichier, puis choisissez Ouvrir.

8. Cliquez sur OK lorsque vous êtes invité à remplacer les paramètres existants et à enregistrer votre nouvelle configuration.

Exemple : configuration SwitchyOmega pour Chrome

L'exemple suivant montre comment configurer l' SwitchyOmegaextension pour Google Chrome. SwitchyOmega vous permet de configurer, de gérer et de basculer entre plusieurs proxys.

Pour installer et configurer à SwitchyOmega l'aide de Google Chrome

1. Accédez à https://chrome.google.com/webstore/category/extensions, recherchez Proxy SwitchyOmega et ajoutez-le à Chrome.
2. Choisissez Nouveau profil et entrez `emr-socks-proxy` comme nom de profil.
3. Choisissez le profil PAC, puis cliquez sur Créer. Les fichiers PAC (configuration automatique de proxy) vous aident à définir une liste d'autorisations pour les requêtes du navigateur qui doivent être transmises à un serveur proxy Web.
4. Dans le champ Script PAC, remplacez le contenu par le script suivant qui définit les URL à transférer via votre serveur proxy Web. Si vous avez spécifié un numéro de port différent lors de la configuration de votre tunnel SSH, remplacez *8157* par votre numéro de port.

```
function FindProxyForURL(url, host) {
    if (shExpMatch(url, "*ec2*.*compute*.amazonaws.com*")) return 'SOCKS5
 localhost:8157';
    if (shExpMatch(url, "*ec2*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "http://10.*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.amazonaws.com*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*.compute.internal*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*ec2.internal*")) return 'SOCKS5 localhost:8157';
    return 'DIRECT';
}
```

5. Sous Actions, choisissez Appliquer les modifications pour enregistrer vos paramètres de proxy.
6. Dans la barre d'outils Chrome, choisissez SwitchyOmega et sélectionnez le `emr-socks-proxy` profil.

Accédez à une interface Web dans le navigateur

Pour ouvrir une interface Web, entrez le nom DNS public de votre nœud principal ou primaire suivi du numéro de port de l'interface choisie dans la barre d'adresse de votre navigateur. L'exemple suivant montre l'URL que vous devez saisir pour vous connecter au Spark HistoryServer.

```
http://master-public-dns-name:18080/
```

Pour obtenir des instructions sur la récupération du nom DNS public d'un nœud, consultez Récupération du nom DNS public du nœud primaire. Pour obtenir la liste complète des URL des interfaces Web, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

Soumission de travail à un cluster

Cette section décrit les méthodes que vous pouvez utiliser pour soumettre du travail à un cluster Amazon EMR. Pour soumettre un travail, vous pouvez ajouter des étapes ou soumettre des tâches Hadoop de manière interactive au nœud primaire.

Tenez compte des règles suivantes relatives au comportement des étapes lorsque vous soumettez des étapes à un cluster :

- Un identifiant d'étape peut contenir jusqu'à 256 caractères.
- Vous pouvez avoir jusqu'à 256 étapes en attente et en cours dans un cluster.
- Même si vous avez 256 étapes actives sur un cluster, vous pouvez soumettre interactivement des travaux au nœud primaire. 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 ou EN ATTENTE à un moment donné.
- Avec Amazon EMR version 4.8.0 et ultérieures, à l'exception de la version 5.0.0, vous pouvez annuler les étapes qui sont en attente. Pour plus d’informations, consultez Annulation d'étapes.
- Avec Amazon EMR versions 5.28.0 et ultérieures, vous pouvez annuler les étapes en attente et actives. Vous pouvez également choisir d'exécuter plusieurs étapes en parallèle pour améliorer l'utilisation du cluster et faire des économies. Pour plus d’informations, consultez Considérations relatives à l'exécution de plusieurs étapes en parallèle.

> **Note**
>
> Pour des performances optimales, nous vous recommandons de stocker les actions d'amorçage personnalisées, les scripts et les autres fichiers que vous souhaitez utiliser avec Amazon EMR dans un compartiment Amazon S3 Région AWS identique à celui de votre cluster.

Rubriques

- Ajouter des étapes à un cluster avec la console de gestion Amazon EMR
- Ajouter des étapes à un cluster à l'aide du AWS CLI
- Considérations relatives à l'exécution de plusieurs étapes en parallèle
- Affichage des étapes
- Annulation d'étapes

Ajouter des étapes à un cluster avec la console de gestion Amazon EMR

Utilisez les procédures suivantes pour ajouter des étapes à un cluster avec la AWS Management Console. Pour obtenir des informations détaillées sur la procédure de soumission des étapes pour des applications Big Data spécifiques, consultez les sections suivantes du Guide de mise à jour d'Amazon EMR :

- Soumission d'une étape JAR personnalisée
- Soumission d'une étape de streaming Hadoop
- Soumission une étape Spark
- Soumission d'une étape Pig
- Exécution d'une commande ou d'un script en tant qu'étape
- Transmission des valeurs aux étapes pour exécuter des scripts Hive

Ajouter des étapes lors de la création du cluster

À partir du AWS Management Console, vous pouvez ajouter des étapes lorsque vous créez un cluster.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Ajouter des étapes lorsque vous créez un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sous Étapes, choisissez Ajouter une étape. Entrez les valeurs appropriées dans les champs de la boîte de dialogue Ajouter une étape. Pour plus d'informations sur le formatage des arguments de vos étapes, consultez Ajouter des arguments d'étape. Les options diffèrent selon le type d'étape. Pour ajouter votre étape et quitter la boîte de dialogue, sélectionnez Ajouter une étape.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer cluster, choisissez Créer un cluster.

Old console

Ajouter des étapes lorsque vous créez un cluster avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/home. Choisissez Créer un cluster : options avancées.
2. Sur la page Step 1: Software and Steps (Étape 1 : Logiciels et étapes) pour Steps (optional) (Étapes (facultatif)), sélectionnez 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 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). Pour plus d'informations sur le formatage des arguments de vos étapes, consultez Ajouter des arguments d'étape. Les options diffèrent selon le type d'étape. Si vous avez activé Exécuter plusieurs étapes en parallèle pour améliorer l'utilisation du cluster et réduire les coûts, la seule option pour Action en cas d'échec est Continuer. Ensuite, choisissez Add (Ajouter).

Ajouter des étapes à un cluster en cours d'exécution

Avec le AWS Management Console, vous pouvez ajouter des étapes à un cluster en désactivant l'option de terminaison automatique.

New console

Ajouter des étapes à un cluster en cours d'exécution à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Étapes de la page de détails du cluster, sélectionnez Ajouter une étape. Pour cloner une étape existante, choisissez le menu déroulant Actions, puis sélectionnez Cloner une étape.
4. Entrez les valeurs appropriées dans les champs de la boîte de dialogue Ajouter une étape. Les options diffèrent selon le type d'étape. Pour ajouter votre étape et quitter la boîte de dialogue, choisissez Ajouter une étape.

Old console

Ajouter des étapes à un cluster en cours d'exécution à l'aide de l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/home. Dans la page Liste de clusters, sélectionnez le lien de 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.

Modifier le niveau de simultanéité des étapes dans un cluster en cours d'exécution

Avec le 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.

New console

Modifier le niveau de simultanéité des étapes dans un cluster en cours d'exécution avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour. Le cluster doit être en cours d'exécution pour modifier son attribut de simultanéité.
3. Dans l'onglet Étapes de la page Détails du cluster, recherchez la section Attributs. Sélectionnez Modifier pour modifier la simultanéité. Saisissez une valeur comprise entre 1 et 256.

Old console

Modifier le niveau de simultanéité des étapes dans un cluster en cours d'exécution avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce/home. Dans la page Liste de clusters, sélectionnez le lien de 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.

Ajouter des arguments d'étape

Lorsque vous utilisez le AWS Management Console pour ajouter une étape à votre cluster, vous pouvez spécifier des arguments pour cette étape dans le champ Arguments. Vous devez séparer les arguments par des espaces et entourer de guillemets les arguments de type chaîne qui sont composés de caractères et d'espaces.

Example : arguments corrects

Les exemples d'arguments suivants sont correctement formatés pour le AWS Management Console, avec des guillemets autour du dernier argument de chaîne.

```
bash -c "aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

Vous pouvez également placer chaque argument sur une ligne distincte pour plus de lisibilité, comme le montre l'exemple suivant.

```
bash
-c
"aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

Example : arguments incorrects

Les exemples d'arguments suivants ne sont pas correctement formatés pour la AWS Management Console. Notez que le dernier argument de chaîne, `aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .`, contient des espaces et n'est pas entouré de guillemets.

```
bash -c aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .
```

Ajouter des étapes à un cluster à l'aide du AWS CLI

Les procédures suivantes montrent comment ajouter des étapes à un cluster nouvellement créé et à un cluster en cours d'exécution à l'aide de l'interface 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. Assurez-vous de remplacer *myKey* par le nom de votre paire de clés Amazon EC2.

```
aws emr create-cluster --name "Test cluster" \
--applications Name=Spark \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-groups InstanceGroupType=PRIMARY,InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

> **Note**
>
> La liste des arguments change en fonction du type d'étape.

Par défaut, le niveau de simultanéité des étapes est 1. Vous pouvez définir le niveau de simultanéité des étapes à l'aide du paramètre `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* par votre propre identifiant de cluster.

```
aws emr add-steps --cluster-id j-2AXXXXXXGAPLF \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

  Le résultat est un identifiant d'étape similaire au suivant.

```
{
    "StepIds": [
 "s-Y9XXXXXXAPMD"
    ]
}
```

Pour modifier le StepConcurrencyLevel dans un cluster en cours d'exécution

1. Dans un cluster en cours d'exécution, vous pouvez le modifier le `StepConcurrencyLevel` à l'aide de l'API `ModifyCluster`. Par exemple, tapez la commande suivante pour augmenter le paramètre le `StepConcurrencyLevel` ou le `10`. Remplacez *j-2AXXXXXXGAPLF* par votre propre identifiant 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 le AWS CLI, consultez la référence des AWS CLI commandes.

Considérations relatives à l'exécution de plusieurs étapes en parallèle

- Les étapes exécutées en parallèle peuvent se terminer dans n'importe quel ordre, mais les étapes en attente passent à l'état d'exécution dans l'ordre dans lequel elles ont été soumises.
- Lorsque vous sélectionnez un niveau de simultanéité d'étapes pour votre cluster, vous devez déterminer si le type d'instance de nœud primaire répond ou non aux exigences en matière de mémoire des charges de travail utilisateur. Le processus d'exécution de l'étape principale s'exécute sur le nœud primaire pour chaque étape. L'exécution de plusieurs étapes en parallèle nécessite plus de mémoire et d'utilisation de l'UC à partir du nœud primaire que l'exécution d'une étape à la fois.
- Pour réaliser une planification complexe et une gestion des ressources d'étapes simultanées, vous pouvez utiliser des fonctions de planification YARN comme `FairScheduler` ou `CapacityScheduler`. Par exemple, vous pouvez utiliser `FairScheduler` avec un ensemble `queueMaxAppsDefault` pour empêcher l'exécution simultanée de plus qu'un certain nombre de tâches.
- Le niveau de simultanéité des étapes est soumis aux configurations des gestionnaires de ressources. Par exemple, si YARN est configuré avec seulement `5` comme valeur de parallélisme, vous ne pouvez avoir que cinq applications YARN s'exécutant en parallèle, même si le paramètres `StepConcurrencyLevel` est défini sur `10`. Pour plus d'informations sur la configuration des gestionnaires de ressources, consultez Configurer les applications dans le Guide de mise à jour Amazon EMR.
- Vous ne pouvez pas ajouter une étape avec un `ActionOnFailure` autre que CONTINUE lorsque le niveau de simultanéité des étapes du cluster est supérieur à 1.
- Si le niveau de simultanéité des étapes d'un cluster est supérieur à 1, la fonctionnalité d'étape `ActionOnFailure` ne sera pas activée.
- Si un cluster possède un niveau de simultanéité d'étapes `1`, mais comporte plusieurs étapes en cours d'exécution, il est possible que `TERMINATE_CLUSTER ActionOnFailure` s'active, mais pas `CANCEL_AND_WAIT ActionOnFailure`. Ce cas limite se présente lorsque le niveau de simultanéité des étapes du cluster était supérieur à 1, mais inférieur à 1 lorsque plusieurs étapes étaient en cours d'exécution.
- Vous pouvez utiliser le dimensionnement automatique EMR pour effectuer une mise à l'échelle vers le haut ou le bas en fonction des ressources YARN afin d'éviter les conflits de ressources. Pour plus d'informations, consultez Utilisation de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances dans le Guide de gestion Amazon EMR.

- Lorsque vous diminuez le niveau de simultanéité des étapes, EMR autorise les étapes en cours d'exécution à se terminer avant de réduire le nombre d'étapes. Si les ressources sont épuisées parce que le cluster exécute trop d'étapes simultanées, nous vous recommandons d'annuler manuellement des étapes en cours d'exécution pour libérer des ressources.

Affichage des étapes

Vous pouvez consulter jusqu'à 10 000 étapes effectuées par Amazon EMR au cours des sept derniers jours. Vous pouvez également consulter les 1 000 étapes effectuées par Amazon EMR à tout moment. Ce total comprend les étapes système et d'utilisateur.

Si vous soumettez de nouvelles étapes une fois que le cluster a atteint la limite d'enregistrement de 1 000 étapes, Amazon EMR supprime les étapes inactives soumises par l'utilisateur dont le statut est TERMINÉ, ANNULÉ ou ÉCHOUÉ depuis plus de sept jours. Si vous soumettez des étapes au-delà de la limite d'enregistrement de 10 000 étapes, Amazon EMR supprime les enregistrements d'étapes inactifs soumis par l'utilisateur, quelle que soit leur durée d'inactivité. Amazon EMR ne supprime pas ces enregistrements des fichiers journaux. Amazon EMR les supprime de la AWS console et ils ne sont pas renvoyés lorsque vous utilisez l'API AWS CLI or pour récupérer les informations du 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
SDK	256 étapes	Jusqu'à 10 000 pas
CLI Amazon EMR	256 étapes	NA
AWS CLI	NA	Jusqu'à 10 000 pas
API	256 étapes	Jusqu'à 10 000 pas

Annulation d'étapes

Vous pouvez annuler les étapes en attente ou en cours depuis l'API AWS Management Console AWS CLI, ou l'API Amazon EMR.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Annuler des étapes avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Étapes de la page de détails du cluster, cochez la case à côté de l'étape que vous souhaitez annuler. Choisissez le menu déroulant Actions, puis sélectionnez Annuler les étapes.
4. Dans la boîte de dialogue Annuler l'étape, choisissez soit d'annuler l'étape et d'attendre qu'elle se termine, soit d'annuler l'étape et de la forcer à se terminer. Ensuite, choisissez Valider.
5. L'état des étapes dans le tableau Étapes devient `CANCELLED`.

Old console

Annuler des étapes avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
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).

CLI

Pour annuler à l'aide du 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-2QUAXXXXXXXXX \
--step-ids s-3M8DXXXXXXXXX s-3M8DXXXXXXXXX \
--step-cancellation-option SEND_INTERRUPT
```

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` : Il s'agit de l'option par défaut. Lorsqu'une demande d'annulation d'étape est reçue, l'EMR envoie un signal `SIGTERM` à l'étape. Ajoutez un gestionnaire de signaux `SIGTERM` à votre logique d'étapes pour capter ce signal et mettre fin aux processus par étapes descendantes, ou attendez qu'ils soient terminés.
- `TERMINATE_PROCESS` : Lorsque cette option est sélectionnée, l'EMR envoie un signal `SIGKILL` à l'étape et à tous ses processus descendants qui les interrompent immédiatement.

Considérations relatives à l'annulation d'étapes

- L'annulation d'une étape en cours ou en attente supprime cette étape du nombre d'étapes actives.
- L'annulation d'une étape en cours ne permet pas à une étape en attente de démarrer, en supposant qu'aucune modification n'a été apportée à `stepConcurrencyLevel`.
- L'annulation d'une étape en cours ne déclenche pas l'étape `ActionOnFailure`.

- Pour EMR 5.32.0 et versions ultérieures, `SEND_INTERRUPT StepCancellationOption` envoie un signal `SIGTERM` au processus enfant de l'étape. Vous devez surveiller ce signal et effectuer un nettoyage et un arrêt en douceur. Le `TERMINATE_PROCESS StepCancellationOption` envoie un signal `SIGKILL` au processus enfant de l'étape et à tous ses processus descendants ; toutefois, les processus asynchrones ne sont pas attribués.

Affichage et surveillance d'un 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 primaire. Vous pouvez également utiliser des services de surveillance tels que CloudWatch 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

- Afficher l'état et les détails d'un cluster
- Amélioration du débogage des étapes
- Afficher l'historique de l'application
- Afficher les fichiers journaux
- Afficher les instances de cluster dans Amazon EC2
- CloudWatch événements et indicateurs
- Affichage des métriques d'application d'un cluster avec Ganglia
- Enregistrement des appels d'API Amazon EMR AWS CloudTrail

Afficher l'état et les détails d'un cluster

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 les métadonnées relatives aux clusters résiliés à titre de référence pendant deux mois, après quoi elles 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.

Pour plus d'informations sur les états du cluster, tels que En attente et En cours d'exécution, consultez Présentation du cycle de vie du cluster.

Afficher les détails du cluster à l'aide de la AWS Management Console

La liste des clusters du site https://console.aws.amazon.com/emr répertorie tous les clusters de votre compte et de votre AWS région, y compris les clusters résiliés. La liste affiche les informations suivantes pour chaque cluster : le nom et l'ID, l'état et les détails de l'état, la date de création, 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.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Afficher les informations du cluster avec la nouvelle console

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

2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez afficher.
3. Utilisez le panneau Récapitulatif pour consulter les éléments de base de la configuration de votre cluster, tels que l'état du cluster, les applications open source installées par Amazon EMR sur le cluster et la version d'Amazon EMR que vous avez utilisée pour créer le cluster. Utilisez chaque onglet sous le Récapitulatif pour afficher les informations, comme décrit dans le tableau suivant.

Old console

Afficher les informations du cluster avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Pour afficher un récapitulatif 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 rattaché à l'étape, explorer les travaux et les tâches de l'étape et accéder aux fichiers journaux.
3. Pour obtenir des informations détaillées sur les clusters, sélectionnez le lien du cluster sous Nom pour ouvrir la page de détails du cluster. Les informations suivantes sont disponibles sur la page de détails du cluster de l'ancienne console :

Onglet (ancienne console)	Description (ancienne console)
Propriétés	Utilisez cet onglet pour consulter le système d'exploitation de votre cluster, les configura tions de terminaison et de sécurité de votre cluster, les informations relatives à votre VPC et à votre sous-réseau, ainsi que l'endroit où vous stockez les journaux dans Amazon S3.
Actions d'amorçage	Utilisez cet onglet pour afficher le statu des actions d'amorçage que le cluster exécute

Onglet (ancienne console)	Description (ancienne console)
	lors de son lancement. Les actions d'amorçage sont utilisées pour les installations logicielles personnalisées et les configurations avancées. Pour plus d'informations, consultez Création d'actions d'amorçage pour installer des logiciels supplémentaires.
Surveillance	Utilisez cet onglet pour afficher les indicateurs clés du fonctionnement du cluster. 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.
Instances	Utilisez cet onglet pour afficher les informations sur les nœuds de votre cluster, y compris les ID des instances EC2, les noms DNS, les volumes EBS et plus encore.
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 Soumission de travail à un cluster.
Applications	Utilisez cet onglet pour afficher les détails d'application persistante du serveur de chronologie YARN et de l'interface utilisateur Tez hors cluster. Vous pouvez également consulter les informations relatives aux applications installées, aux configurations de clusters et aux groupes d'instances. Les interfaces utilisateur d'application sur cluster sont disponibles pendant l'exécution du cluster.

Onglet (ancienne console)	Description (ancienne console)
Evénements	Utilisez cet onglet pour afficher le journal des événements de votre cluster. Pour plus d'informations, consultez Surveillance des événements Amazon EMR avec CloudWatch.
Balises	Utilisez cet onglet pour afficher les balises que vous avez appliquées au cluster.

Consultez les détails du cluster à l'aide du 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 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. Pour de plus amples informations sur les états des clusters, consultez Présentation du cycle de vie du cluster. L'exemple suivant illustre l'utilisation de la commande `describe-cluster`, suivie par des exemples de la commande list-clusters.

Example Affichage de l'état d'un 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 incluent : DÉMARRAGE EN COURS, ACTION D'AMORÇAGE, EN COURS D'EXÉCUTION, EN ATTENTE, TERMINÉ et TERMINÉ AVEC DES ERREURS.

```
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` filtre les clusters dont l'état est STARTING, BOOTSTRAPPING, RUNNING, WAITING ou TERMINATING.
- `--terminated` filtre les clusters ayant l'état TERMINATED.
- Le paramètre `--failed` filtre les clusters ayant l'état TERMINATED_WITH_ERRORS.

Les commandes suivantes retournent le même résultat.

```
aws emr list-clusters --cluster-states TERMINATED
```

```
aws emr list-clusters --terminated
```

Pour de plus amples informations sur les états des clusters, consultez Présentation du cycle de vie du cluster.

Amélioration du débogage des étapes

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 :

- Une erreur courante Hadoop, par exemple lorsque le répertoire de sortie existe déjà, que le répertoire d'entrée n'existe pas ou que la mémoire est insuffisante pour une application.
- Des erreurs Java, par exemple une application compilée avec une version incompatible de Java ou exécutée avec une classe principale introuvable.
- Un problème pour accéder aux objets stockés dans Amazon S3.

Ces informations sont disponibles à l'aide des opérations DescribeStepet de ListStepsl'API. Le FailureDetailschamp des informations StepSummaryrenvoyées par ces opérations. Pour accéder aux FailureDetails informations, utilisez la AWS CLI, la console ou le AWS SDK.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

La nouvelle console Amazon EMR ne propose pas de débogage par étapes. Cependant, vous pouvez consulter les détails de la terminaison du cluster en suivant les étapes suivantes.

Afficher les informations relatives aux échecs à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez afficher.
3. Notez la valeur Statut dans la section Récapitulatif de la page de détails du cluster. Si l'état est Terminé avec des erreurs, passez le curseur sur le texte pour afficher les détails de la défaillance du cluster.

Old console

Afficher les informations relatives aux échecs à l'aide de l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Liste de clusters 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.

CLI

Pour consulter les détails de la panne à l'aide du AWS CLI

- Pour obtenir les détails de l'échec d'une étape avec le AWS CLI, utilisez la `describe-step` commande.

```
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 consulter les détails d'application du serveur d'historique Spark et du service de chronologie YARN sur la page de détails du cluster dans la console. L'historique d'application Amazon EMR facilite la résolution des problèmes et l'analyse des travaux actifs et de l'historique des travaux.

> **Note**
>
> Pour renforcer la sécurité des applications hors console que vous pouvez utiliser avec Amazon EMR, les domaines hébergeant les applications sont enregistrés dans la liste des suffixes publics (PSL). Voici des exemples de ces domaines d'hébergement : `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Pour plus de sécurité, si vous avez besoin de définir des cookies sensibles dans le nom de domaine par défaut, nous vous recommandons d'utiliser des cookies avec un préfixe `__Host-`. Cela vous permettra de protéger votre domaine contre les tentatives de falsification de requêtes intersites (CSRF). Pour plus d'informations, voir la page Set-Cookie du Mozilla Developer Network.

La section Interfaces utilisateur des applications de l'onglet Applications propose plusieurs options d'affichage, en fonction de l'état du cluster et des applications que vous avez installées sur le cluster.

- Accès hors cluster aux interfaces utilisateur d'application persistante : À partir de la version 5.25.0 d'Amazon EMR, des liens d'interface utilisateur d'application persistante sont disponibles pour l'interface utilisateur Spark et le serveur d'historique Spark. Avec la version 5.30.1 et ultérieure d'Amazon EMR, l'interface utilisateur Tez et le serveur de chronologie YARN disposent 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. Pour plus d'informations et d'instructions sur la configuration d'un tunnel SSH, consultez Affichage des interfaces Web hébergées sur des clusters Amazon EMR.

  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.

Afficher les interfaces utilisateur d'application persistante

À partir de la version 5.25.0 d'Amazon EMR, vous pouvez vous connecter aux détails de l'application persistante de serveur d'historique Spark hébergée hors cluster à l'aide de la page Récapitulatif du cluster ou de l'onglet Interfaces utilisateur d'application de la console. Les interfaces d'application persistante de l'interface utilisateur Tez et du serveur de chronologie YARN sont disponibles à partir de la version 5.30.1 d'Amazon EMR. 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.

Accédez aux détails de votre cluster dans la console, puis sélectionnez l’onglet Applications. Sélectionnez l’interface utilisateur de l’application souhaitée une fois votre cluster lancé. L’interface utilisateur de l’application s’ouvre dans un nouvel onglet de navigateur. Pour plus d'informations, consultez Surveillance et 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 vous n'activez pas la journalisation, 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 :

- Les journaux 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 politique Amazon S3 pour le sous-réseau privé. Pour plus d'informations, consultez Politique Amazon S3 minimale pour un sous-réseau privé.
- Les journaux 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, consultez 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-7.1.0 \
--applications Name=Hadoop Name=Spark --ec2-attributes KeyName=<myEMRKeyPairName> \
--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://region.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.

Fichiers journaux d'événements Spark volumineux

Dans certains cas, les tâches Spark de longue durée, telles que le streaming Spark, et les tâches volumineuses, telles que les requêtes SQL Spark, peuvent générer des journaux d'événements volumineux. Avec des journaux d'événements volumineux, l'espace disque des instances de calcul peut être rapidement utilisé et des erreurs `OutOfMemory` peuvent survenir lors du chargement des interfaces utilisateur persistantes. Pour éviter ces problèmes, nous vous recommandons d'activer la fonctionnalité de roulement et de compactage du journal des événements de Spark. Cette fonctionnalité est disponible sur Amazon EMR versions emr-6.1.0 et ultérieures. Pour plus de détails sur le laminage et le compactage, consultez Appliquer le compactage aux fichiers journaux des événements de propagation dans la documentation de Spark.

Pour activer la fonctionnalité de propagation et de compactage du journal des événements Spark, activez les paramètres de configuration Spark suivants.

- `spark.eventLog.rolling.enabled` : Active la propagation du journal des événements en fonction de la taille. Ce paramètre est désactivé par défaut.
- `spark.eventLog.rolling.maxFileSize` : Lorsque la propagation est activée, spécifie la taille maximale du fichier journal des événements avant qu'il ne soit reporté. La valeur par défaut est 128 Mo.
- `spark.history.fs.eventLog.rolling.maxFilesToRetain` : Spécifie le nombre maximal de fichiers journaux d'événements non compactés à retenir. Par défaut, tous les fichiers journaux d'événements sont retenus. Réglez sur une valeur inférieure pour compacter les anciens journaux d'événements. La valeur la plus faible est 1.

Notez que le compactage tente d'exclure les événements dont les fichiers journaux d'événements sont obsolètes, tels que les suivants. S'il supprime des événements, vous ne les verrez plus dans l'interface utilisateur du serveur d'historique Spark.

- Événements relatifs aux tâches terminées et événements liés à une étape ou à une tâche.
- Événements pour les exécuteurs suspendus.

- Événements relatifs aux requêtes SQL terminées et aux événements relatifs aux tâches, aux étapes et aux tâches connexes.

Lancer un cluster avec la propagation et le compactage activés

1. Créez un fichier `spark-configuration.json` avec la configuration suivante.

```
[
   {
     "Classification": "spark-defaults",
        "Properties": {
           "spark.eventLog.rolling.enabled": true,
           "spark.history.fs.eventLog.rolling.maxFilesToRetain": 1
        }
   }
]
```

2. Créez votre cluster avec la configuration de compactage par propagation de Spark comme suit.

```
aws emr create-cluster \
--release-label emr-6.6.0 \
--instance-type m4.large \
--instance-count 2 \
--use-default-roles \
--configurations file://spark-configuration.json
```

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 dans 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 fonctionnalité 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 Amazon 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 chacune d'entre elles Compte AWS, la limite par défaut pour les interfaces utilisateur d'applications actives est de 200.
- Dans ce qui suit Régions AWS, vous pouvez accéder aux interfaces utilisateur des applications depuis la console avec Amazon EMR 6.14.0 ou version ultérieure :
  - Asie-Pacifique (Jakarta) (ap-southeast-3)
  - Europe (Espagne) (eu-south-2)
  - Asie-Pacifique (Melbourne) (ap-southeast-4)
  - Israël (Tel Aviv) (il-central-1)
  - Moyen-Orient (Émirats arabes unis) (me-central-1)
- Dans ce qui suit Régions AWS, vous pouvez accéder aux interfaces utilisateur des applications depuis la console avec Amazon EMR 5.25.0 ou version ultérieure :
  - USA Est (Virginie du Nord) (us-east-1)
  - USA Ouest (Oregon) (us-west-2)
  - Asie-Pacifique (Mumbai) (ap-south-1)
  - Asie-Pacifique (Séoul) (ap-northeast-2)
  - Asie-Pacifique (Singapour) (ap-southeast-1)
  - Asie-Pacifique (Sydney) (ap-southeast-2)
  - Asie-Pacifique (Tokyo) (ap-northeast-1)
  - Canada (Centre) (ca-central-1)
  - Amérique du Sud (São Paulo) (sa-east-1)
  - Europe (Francfort) (eu-central-1)
  - Europe (Irlande) (eu-west-1)
  - Europe (Londres) (eu-west-2)
  - Europe (Paris) (eu-west-3)

- Europe (Stockholm) (eu-north-1)
- Chine (Beijing) cn-north-1
- Chine (Ningxia) cn-northwest-1

Afficher un historique détaillé des applications

> **Note**
>
> Pour une expérience utilisateur améliorée qui conserve l'historique pendant 30 jours maximum, nous vous recommandons d'utiliser l'interface de l'application persistante. L'historique détaillé des applications décrit sur cette page n'est pas disponible dans la nouvelle console Amazon EMR (https://console.aws.amazon.com/emr). Pour plus d'informations, consultez Afficher les interfaces utilisateur d'application persistante.

Avec les versions 5.8.0 à 5.36.0 d'Amazon EMR et les versions 6.x jusqu'à 6.8.0, vous pouvez consulter un historique détaillé des applications depuis l'onglet Interfaces utilisateur des applications de l'ancienne console Amazon EMR. L'interface utilisateur des applications Amazon EMR conserve le récapitulatif de l'historique des applications pendant 7 jours après leur finalisation.

Considérations et restrictions

Tenez compte des limites suivantes lorsque vous utilisez l'onglet Interfaces utilisateur d'application dans l'ancienne console Amazon EMR.

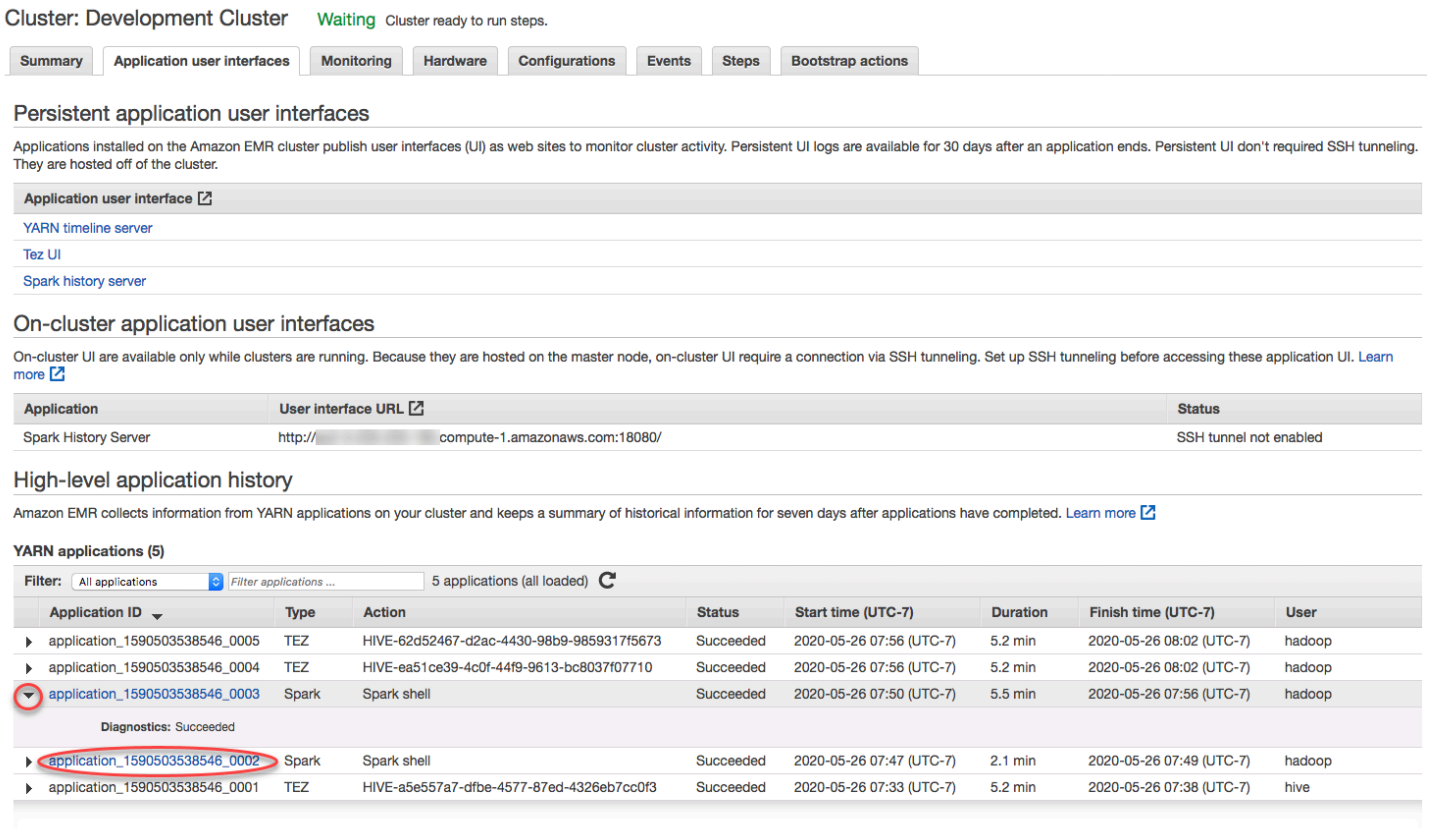
- Vous ne pouvez accéder à la fonctionnalité d'historique des applications de haut niveau que lorsque vous utilisez les versions 5.8.0 à 5.36.0 d'Amazon EMR et les versions 6.x jusqu'à 6.8.0. À compter du 23 janvier 2023, Amazon EMR mettra fin à l'historique détaillé des applications pour toutes les versions. Si vous utilisez Amazon EMR version 5.25.0 ou supérieure, nous vous recommandons d'utiliser plutôt l'interface utilisateur d'application persistante.
- La fonctionnalité d'historique des applications de haut niveau ne prend pas en charge les applications Spark Streaming.
- L'accès en un clic aux interfaces utilisateur d'application persistante n'est actuellement pas disponible pour les clusters Amazon EMR dotés de plusieurs nœuds principaux ou pour les clusters Amazon EMR intégrés à AWS Lake Formation.

Exemple : Afficher un historique détaillé des applications

La séquence suivante illustre l'exploration des détails des tâches d'une application Spark ou YARN via l'onglet Interfaces utilisateur des applications sur la page de détails du cluster de l'ancienne console.

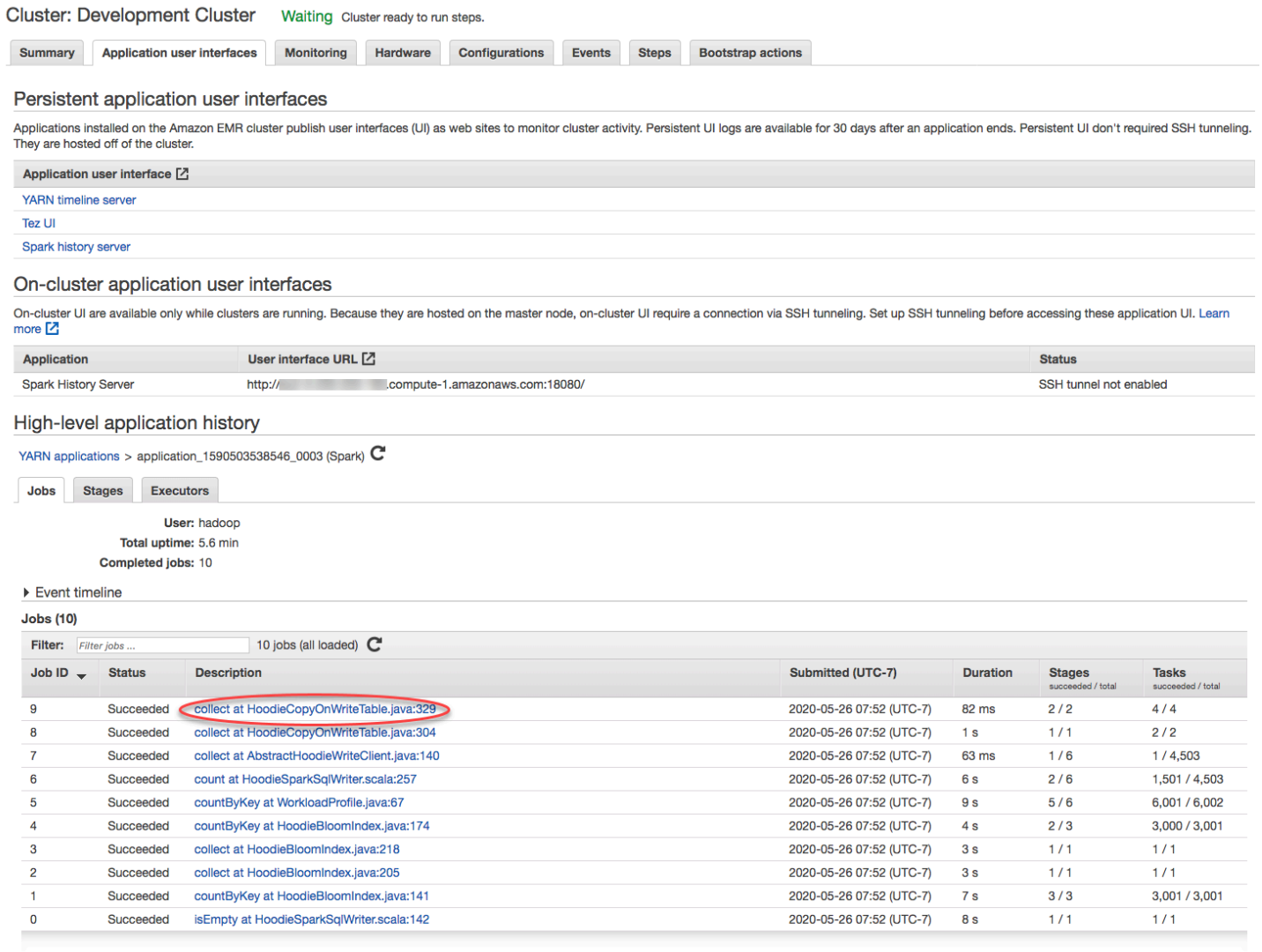
Pour afficher les détails d'un cluster, sélectionnez un nom de cluster dans la liste Clusters. Pour afficher des informations sur les journaux de conteneur YARN, vous devez activer la journalisation pour votre cluster. Pour plus d'informations, consultez Configuration de la journalisation et du débogage 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'onglet Interfaces utilisateur des applications, sous Historique des applications de haut niveau, vous pouvez développer une ligne pour afficher le récapitulatif des diagnostics d'une application Spark ou sélectionner un lien d'ID d'application pour afficher les détails d'une autre application.

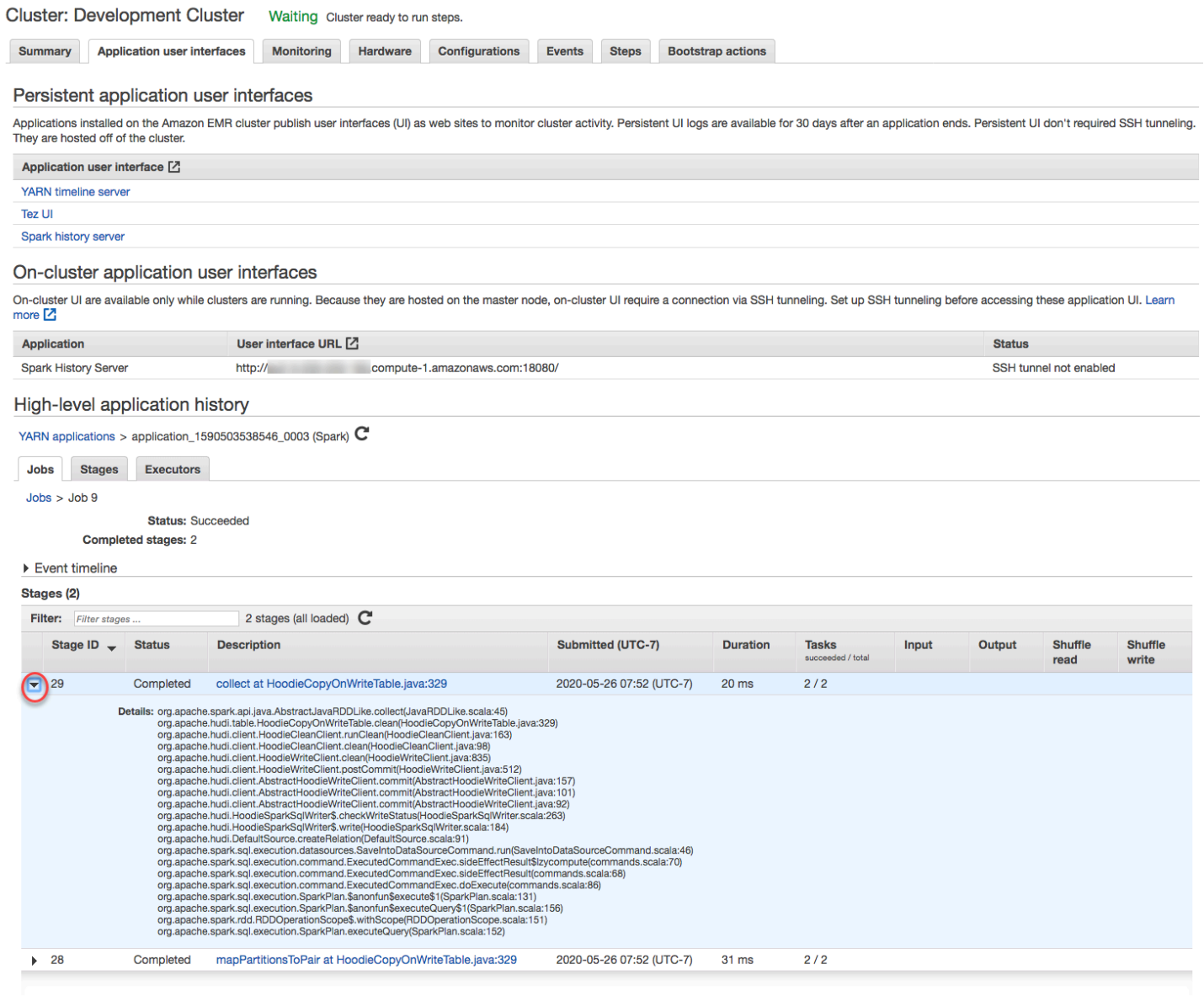


Lorsque vous sélectionnez un lien d'ID d'application, l'interface utilisateur change pour afficher les détails de l'application YARN pour cette application. Dans l'onglet Tâches des détails de

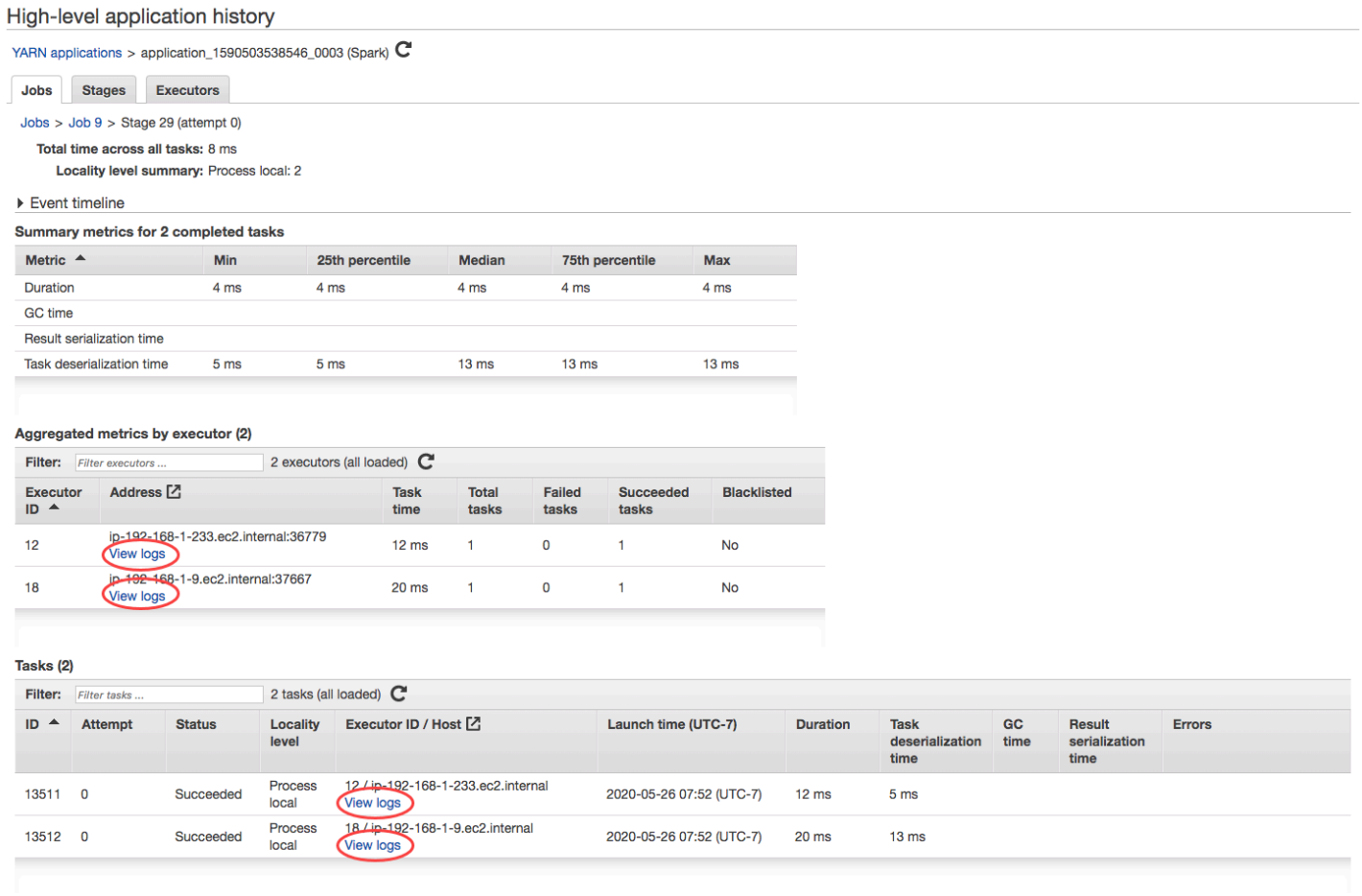
l'application YARN, vous pouvez choisir le lien Description d'une tâche pour afficher les détails de cette tâche.



Sur la page des détails de la tâche, vous pouvez développer des informations sur les différentes étapes de la tâche, puis sélectionner le lien Description pour voir les détails de l'étape.



Sur la page des détails de l'étape, vous pouvez consulter les indicateurs clés des tâches de l'étape et des exécuteurs. Vous pouvez également consulter les journaux des tâches et des exécuteurs à l'aide des liens Afficher les journaux.



Afficher les fichiers journaux

Amazon EMR et Hadoop génèrent des fichiers journaux qui indiquent l'état du cluster. Par défaut, ces journaux sont écrits sur le nœud primaire 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 primaire. Amazon EMR écrit les journaux des étapes, des actions de démarrage et de l'état des 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, rendez-vous sur http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup .html.

Affichage des fichiers journaux sur le nœud primaire

Le tableau suivant répertorie quelques-uns des fichiers journaux que vous trouverez sur le nœud primaire.

Emplacement	Description
/emr/instance-controller/log/bootstrap-actions	Journaux écrits pendant le traitement des actions amorçage.
/mnt/var/log/hadoop-state-pusher	Journaux écrits par le processus de transmiss ion d'état Hadoop.
/emr/instance-controller/log	Journaux de contrôleur d'instance.
/emr/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.
/emr/service-nanny	Journaux écrits par le processus de surveilla nce 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 informati ons sur le traitement de l'étape. La valeur *N* indique la valeur stepId attribuée 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 dans `/mnt/var/log/hadoop/steps/s-5678IJKLMNOP/` . Les journaux d'étapes rédigés par Amazon EMR sont les suivants.

Emplacement	Description
	• Contrôleur : Informations sur le traitement de l'étape. Si votre étape échoue lors du chargement, vous pouvez trouver la trace de la pile dans ce journal. • syslog : Décrit l'exécution des tâches Hadoop au cours de l'étape. • stderr : Le canal d'erreur standard de Hadoop pendant le traitement de l'étape. • stdout : Le canal de sortie standard de Hadoop pendant le traitement de l'étape.

Afficher des fichiers journaux sur le nœud primaire avec l'interface AWS CLI.

1. Utilisez le protocole SSH pour vous connecter au nœud primaire, comme décrit dans Connexion au nœud primaire à l'aide de SSH.
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
   ```

Afficher des fichiers journaux archivés dans 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.

Lorsqu'Amazon EMR est configuré pour archiver les fichiers journaux dans Amazon S3, il stocke les fichiers dans l'emplacement S3 spécifié, dans le dossier /*cluster-id*/, où *cluster-id* est l'identifiant du cluster.

Le tableau suivant répertorie quelques-uns des fichiers journaux disponibles dans Amazon S3.

Emplacement	Description
/*cluster-id* /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.
/*cluster-id* /node/*instance-id* /*application*	Journaux créés par chaque application ou démon rattaché à une application. Par exemple, le journal du serveur Hive est situé dans *cluster-id* /node/*instance-id* /hive/hive-server.log .
/*cluster-id* /steps/*step-id*/	Journaux d'étape qui contiennent des informations sur le traitement de l'étape. La valeur de *step-id* indique l'identifiant d'étape 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 dans /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/ . Les journaux d'étapes rédigés par Amazon EMR sont les suivants. • Contrôleur : Informations sur le traitement de l'étape. Si votre étape échoue lors du

Emplacement	Description
	chargement, vous pouvez trouver la trace de la pile dans ce journal. • syslog : Décrit l'exécution des tâches Hadoop au cours de l'étape. • stderr : Le canal d'erreur standard de Hadoop pendant le traitement de l'étape. • stdout : Le canal de sortie standard de Hadoop pendant le traitement de l'étape.
/*cluster-id* /containers	Journaux de conteneur d'applications. Les journaux pour chaque application YARN sont stockés à ces emplacements.
/*cluster-id* /hadoop-mapreduce/	Les journaux qui contiennent des informations sur les détails de configuration et l'historique des MapReduce tâches.

Afficher les fichiers journaux archivés sur Amazon S3 à l'aide de la console Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la 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. Téléchargez l'objet de fichier journal pour l'afficher. Pour obtenir des instructions, consultez Téléchargement d'un objet.

Afficher des fichiers journaux dans l'outil de débogage

Amazon EMR n'active pas automatiquement l'outil de débogage. Vous devez configurer celui-ci lorsque vous lancez le cluster. Notez que la nouvelle console Amazon EMR ne propose pas d'outil de débogage.

Afficher les journaux du cluster avec l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Sur la page Liste des clusters, cliquez sur l'icône des détails à côté du cluster que vous souhaitez afficher.

   La page de détails du cluster s'ouvre. Dans la section Étapes, 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 tâches Hadoop rattachées à une étape donnée, cliquez sur le lien Afficher les tâches à droite de l'étape.
4. Pour afficher la liste des tâches Hadoop rattachées à une tâche donnée, cliquez sur le lien Afficher les tâches à droite de la tâche.
5. Pour consulter la liste des tentatives effectuées par une tâche donnée pendant qu'elle tentait de terminer, cliquez sur le lien Afficher les tentatives situé à droite de la tâche.
6. Pour consulter les journaux générés par une tentative de tâche, choisissez 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 Actualiser la liste dans les volets de débogage pour obtenir le meilleur up-to-date état de ces éléments.

Afficher les instances de cluster dans Amazon EC2

Pour vous aider à gérer vos ressources, Amazon EC2 vous permet d'attribuer 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 que vous attribuez aux ressources par le biais de votre AWS compte ne sont disponibles que pour vous. Les autres comptes qui partagent la même ressource ne peuvent pas afficher vos balises..

Amazon EMR étiquette automatiquement chaque instance EC2 qu'il lance avec des paires clé-valeur. Les clés identifient le cluster et le groupe d'instances auxquels appartient l'instance. 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 d'instance pour la tâche. 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	Valeur	Valeur : Définition
aws:elasticmapreduce:job-flow-id	*job-flow-identifier*	ID du cluster pour lequel l'instance est mise en service. Il apparaît dans le format `j-XXXXXXXXXXXXX` suivant et peut comporter jusqu'à 256 caractères.
aws:elasticmapreduce:instance-group-role	*group-role*	Type de groupe d'instances, saisi sous la forme de l'une des valeurs suivantes : `master`, `core` `outask`.

Vous pouvez afficher et filtrer sur les balises qu'Amazon EMR ajoute. Pour plus d'informations, consultez la section Utilisation des balises dans le guide de l'utilisateur Amazon EC2. Etant donné que les balises définies par Amazon EMR sont des balises système et qu'elles ne peuvent pas être modifiées ou supprimées, les sections sur les balises 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. En cas de latence entre le moment où l'instance EC2 est mise en service et le moment où son statut est défini sur En cours d'exécution, les balises définies par Amazon EMR apparaîtront une fois l'instance démarrée. Si vous ne voyez pas les balises, attendez quelques minutes et actualisez la vue.

CloudWatch événements et indicateurs

Utilisez des événements ainsi que des métriques pour suivre l'activité et l'état de santé d'un cluster Amazon EMR. 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 plus d'informations sur les CloudWatch événements, consultez le guide de l'utilisateur Amazon CloudWatch Events. Pour plus d'informations sur CloudWatch les métriques, consultez les sections Utilisation CloudWatch des métriques Amazon et Création d' CloudWatchalarmes Amazon dans le guide de CloudWatch l'utilisateur Amazon.

Rubriques

- Surveillance des métriques Amazon EMR avec CloudWatch
- Surveillance des événements Amazon EMR avec CloudWatch
- Réagir aux CloudWatch événements

Surveillance des métriques Amazon EMR avec CloudWatch

Les métriques sont mises à jour toutes les cinq minutes et automatiquement collectées et transmises CloudWatch pour chaque cluster Amazon EMR. Cet intervalle n'est pas configurable. Les métriques Amazon EMR indiquées dans le document sont gratuites. CloudWatch Ces métriques de points de données de cinq minutes sont archivées pendant 63 jours, après quoi les données sont supprimées.

Comment utiliser les métriques Amazon EMR ?

Le tableau suivant montre les utilisations courantes des mesures signalées par Amazon EMR. Voici quelques suggestions pour vous aider à démarrer, qui ne forment pas une liste exhaustive. Pour consulter une liste complète des métriques présentées par Amazon EMR, consultez Métriques rapportées par Amazon EMR dans CloudWatch.

Comment... ?	Métriques pertinentes
Suivre la progression de mon cluster	Prenez en compte les métriques `RunningMapTasks` , `RemainingMapTasks` ,

Comment... ?	Métriques pertinentes
	`RunningReduceTasks` et `Remaining ReduceTasks` .
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 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 `MRUnhealthyNodes` suit le moment où un ou plusieurs nœuds principau x ou de tâches sont à court de stockage sur disque local et passent à l'état YARN `UNHEALTHY` . Par exemple, les nœuds principaux ou les nœuds de tâches manquent d'espace disque et ne seront pas en mesure d'exécuter des tâches.
Détecter lorsqu'un cluster ne dispose plus d'un espace de stockage suffisant	La métrique `HDFSUtilization` surveille la capacité HDFS combinée du cluster et peut nécessiter le redimensionnement du cluster pour ajouter d'autres nœuds principaux. Par exemple, l'utilisation du HDFS est élevée, ce qui peut attribuer les tâches et l'état du cluster.
Détecter lorsqu'un cluster fonctionne à capacité réduite	La métrique `MRLostNodes` surveille les problèmes de communication de plusieurs nœuds principaux ou de tâches avec le nœud principal. Par exemple, le nœud principal ou le nœud de tâches n'est pas accessible par le nœud principal.

Pour plus d'informations, consultez Le cluster se termine avec NO_SLAVE_LEFT et les nœuds principaux FAILED_BY_MASTER. et AWSSupport-AnalyzeEMRLogs.

CloudWatch Mesures d'accès pour Amazon EMR

Vous pouvez consulter les métriques transmises par Amazon EMR à CloudWatch l'aide de la console Amazon EMR ou de la console. CloudWatch Vous pouvez également récupérer des métriques à l'aide de la commande CloudWatch CLI `mon-get-stats` ou de l'CloudWatch `GetMetricStatistics`API. Pour plus d'informations sur l'affichage ou la récupération de métriques pour Amazon EMR à l' CloudWatchaide d'Amazon EMR, consultez le guide de l'utilisateur CloudWatch Amazon.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Consulter les métriques avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis le cluster pour lequel vous souhaitez consulter les métriques. La page de détails du cluster s'ouvre.
3. Sélectionnez l'onglet Surveillance sur la page de détails du cluster. Pour charger les rapports sur la progression et l'état du cluster, choisissez l'une des options suivantes : État du cluster, État du nœud ou Entrées et sorties.
4. Une fois que vous avez choisi une métrique à afficher, vous pouvez agrandir chaque graphique. Pour filtrer la période de votre graphique, sélectionnez une option préremplie ou choisissez Personnalisé.

Old console

Consulter les métriques avec l'ancienne console

1. Ouvrez la console Amazon EMR à l'adressehttps://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 intitulés État du cluster, Mapper/Réduire, État du nœud ou E/S pour charger les rapports sur la progression et l'état de santé 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.

Métriques rapportées par Amazon EMR dans CloudWatch

Les tableaux suivants répertorient les métriques qu'Amazon EMR rapporte dans la console et vers lesquelles il envoie des messages. CloudWatch

Métriques Amazon EMR

Amazon EMR envoie les données de plusieurs métriques à. 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

Métrique	Description
	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 : surveiller les performances du cluster Unités : booléennes
ContainerAllocated	Le nombre de conteneurs de ressources alloués par leResourceManager. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
ContainerReserved	Nombre de conteneurs réservés. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
ContainerPending	Nombre de conteneurs dans la file d'attente qui n'ont pas encore été alloués. Cas d'utilisation : surveiller la progression du cluster Unités : nombre

Métrique	Description
ContainerPendingRatio	Le rapport entre les conteneurs en attente et les conteneurs alloués (ContainerPendingRatio = ContainerPending / ContainerAllocated). Si ContainerAllocated = 0, alors ContainerPendingRatio =ContainerPending. La valeur de ContainerPendingRatio représente un nombre et non un pourcentage. Cette valeur est utile pour dimensionner les ressources de cluster en fonction du comportement d'attribution des conteneurs. Unités : nombre
AppsCompleted	Nombre de demandes soumises à YARN ayant été traitées. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
AppsFailed	Nombre de demandes soumises à YARN impossibles à traiter. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre
AppsKilled	Nombre d'applications soumises à YARN ayant été désactivées. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre

Métrique	Description
AppsPending	Nombre d'applications soumises à YARN qui se trouvent dans un état d'attente. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
AppsRunning	Nombre d'applications soumises à YARN qui sont en cours d'exécution. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
AppsSubmitted	Nombre d'applications soumises à YARN. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
Statut du nœud	
CoreNodesCourir	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 : surveiller l'intégrité du cluster Unités : nombre

Métrique	Description
CoreNodesEn attente	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 : surveiller l'intégrité du cluster Unités : nombre
LiveDataNœuds	Pourcentage de nœuds de données qui reçoivent des tâches de Hadoop. Cas d'utilisation : surveiller l'intégrité du cluster Unités : pourcentage
M. TotalNodes	Le nombre de nœuds actuellement disponibles pour les MapReduce tâches. Équivalent à la métrique YARN `mapred.resourcemanager.TotalNodes` . Cas d'utilisation : surveiller la progression du cluster Unités : nombre
M. ActiveNodes	Le nombre de nœuds exécutant actuellement MapReduce des tâches ou des tâches. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfActiveNodes` . Cas d'utilisation : surveiller la progression du cluster Unités : nombre

Métrique	Description
M. LostNodes	Le nombre de nœuds alloués MapReduce qui ont été marqués comme étant PERDUS. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfLostNodes` . Cas d'utilisation : surveiller l'intégrité du cluster, surveiller la progression du cluster Unités : nombre
M. UnhealthyNodes	Le nombre de nœuds disponibles pour les MapReduce tâches marquées dans un état INSTABLE. Équivalent à la métrique YARN `mapred.resourcemanager.NoOf UnhealthyNodes` . Cas d'utilisation : surveiller la progression du cluster Unités : nombre
M. DecommissionedNodes	Le nombre de nœuds alloués aux MapReduce applicati ons qui ont été marquées comme étant HORS SERVICE. Équivalent à la métrique YARN `mapred.resourceman ager.NoOfDecommissionedNodes` . Cas d'utilisation : surveiller l'intégrité du cluster, surveiller la progression du cluster Unités : nombre
M. RebootedNodes	Le nombre de nœuds disponibles MapReduce qui ont été redémarrés et marqués dans l'état REDÉMARRÉ. Équivalent à la métrique YARN `mapred.resourceman ager.NoOfRebootedNodes` . Cas d'utilisation : surveiller l'intégrité du cluster, surveiller la progression du cluster Unités : nombre

Métrique	Description
MultiMasterInstanceGroupNodesRunning	Le nombre de nœuds maîtres en cours d'exécution. Cas d'utilisation : surveiller l'échec et le remplacement du nœud maître Unités : nombre
MultiMasterInstanceGroupNodesRunningPourcentage	Le pourcentage de nœuds principaux en cours d'exécution sur le nombre d'instances de nœuds principaux demandées. Cas d'utilisation : surveiller l'échec et le remplacement du nœud maître Unités : pourcentage
MultiMasterInstanceGroupNodesRequested	Le nombre de nœuds maîtres demandés. Cas d'utilisation : surveiller l'échec et le remplacement du nœud maître Unités : nombre
E/S	
S3 BytesWritten	Nombre d'octets écrits sur Amazon S3. Cette métrique regroupe uniquement les MapReduce tâches et ne s'applique pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre

Métrique	Description
S3 BytesRead	Nombre d'octets lus à partir d'Amazon S3. Cette métrique regroupe uniquement les MapReduce tâches et ne s'appliqu e pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
HDFSUtilization	Pourcentage de stockage HDFS actuellement utilisé. Cas d'utilisation : analyser les performances du cluster Unités : pourcentage
HDFS BytesRead	Nombre d'octets lus à partir de HDFS. Cette métrique regroupe uniquement les MapReduce tâches et ne s'appliqu e pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
HDFS BytesWritten	Nombre d'octets écrits sur HDFS. Cette métrique regroupe uniquement les MapReduce tâches et ne s'applique pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
MissingBlocks	Nombre de blocs dans lesquels HDFS n'a aucun réplica. Il peut s'agir de blocs corrompus. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre

Métrique	Description
CorruptBlocks	Nombre de blocs que HDFS indique comme étant corrompus. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre
TotalLoad	Nombre total de transferts de données simultanés. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre
MemoryTotalMB	Quantité totale de mémoire dans le cluster. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
MemoryReservedMB	Quantité de mémoire réservée. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
MemoryAvailableMB	Quantité de mémoire disponible à allouer. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
MemoryAvailablePourcentage de fil	Pourcentage de mémoire restante disponible pour YARN (YARN MemoryAvailablePercentage = MemoryAvailable MemoryTotal Mo/Mo). Cette valeur est utile pour dimension ner les ressources de cluster en fonction de l'utilisation de mémoire YARN. Unités : pourcentage

Métrique	Description
MemoryAllocatedMB	Quantité de mémoire allouée au cluster. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
PendingDeletionBlocs	Nombre de blocs marqués pour la suppression. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre
UnderReplicatedBlocs	Nombre de blocs devant être répliqués une ou plusieurs fois. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre
DfsPendingReplicationBlocks	État de la réplication des blocs : blocs en cours de réplicati on, l'âge des demandes de réplication et demandes de réplication ayant échoué. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre
CapacityRemainingGo	Quantité de capacité du disque HDFS restante. Cas d'utilisation : surveiller la progression du cluster, surveiller l'intégrité du cluster Unités : nombre

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 : surveiller les performances du cluster Unités : booléennes
JobsRunning	Nombre de tâches en cours d'exécution dans le cluster. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre
JobsFailed	Nombre de tâches qui ont échoué dans le cluster. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre
Mapper/Réduire	
MapTasksCourir	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.

Métrique	Description
	Cas d'utilisation : surveiller la progression du cluster Unités : nombre
MapTasksRestant	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 correspond à une tâche dont l'état n'est pas l'un des états suivants : en cours d'exécution, désactivé ou terminé. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
MapSlotsOuvert	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 les performances du cluster Unités : nombre
RemainingMapTasksPerFente	Rapport entre les tâches de mappage total restantes et le nombre total d'emplacements de mappage disponibles dans le cluster. Cas d'utilisation : analyser les performances du cluster Unités : rapport

Métrique	Description
ReduceTasksCourir	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 : surveiller la progression du cluster Unités : nombre
ReduceTasksRestant	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 : surveiller la progression du cluster Unités : nombre
ReduceSlotsOuvert	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 les performances du cluster Unités : nombre
Statut du nœud	
CoreNodesCourir	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 : surveiller l'intégrité du cluster Unités : nombre

Métrique	Description
CoreNodesEn attente	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 : surveiller l'intégrité du cluster Unités : nombre
LiveDataNœuds	Pourcentage de nœuds de données qui reçoivent des tâches de Hadoop. Cas d'utilisation : surveiller l'intégrité du cluster Unités : pourcentage
TaskNodesCourir	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 : surveiller l'intégrité du cluster Unités : nombre
TaskNodesEn attente	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 : surveiller l'intégrité du cluster Unités : nombre

Métrique	Description
LiveTaskTraceurs	Pourcentage de dispositifs de suivi des tâches fonctionnels. Cas d'utilisation : surveiller l'intégrité du cluster Unités : pourcentage
E/S	
S3 BytesWritten	Nombre d'octets écrits sur Amazon S3. Cette métrique regroupe uniquement les MapReduce tâches et ne s'appliqu e pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
S3 BytesRead	Nombre d'octets lus à partir d'Amazon S3. Cette métrique regroupe uniquement les MapReduce tâches et ne s'appliqu e pas aux autres charges de travail sur Amazon EMR. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
HDFSUtilization	Pourcentage de stockage HDFS actuellement utilisé. Cas d'utilisation : analyser les performances du cluster Unités : pourcentage
HDFS BytesRead	Nombre d'octets lus à partir de HDFS. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre

Métrique	Description
HDFS BytesWritten	Nombre d'octets écrits sur HDFS. Cas d'utilisation : analyser les performances du cluster, surveiller la progression du cluster Unités : nombre
MissingBlocks	Nombre de blocs dans lesquels HDFS n'a aucun réplica. Il peut s'agir de blocs corrompus. Cas d'utilisation : surveiller l'intégrité du cluster Unités : nombre
TotalLoad	Le nombre total actuel de lecteurs et d'écrivains déclarés par tous les membres DataNodes d'un cluster. Cas d'utilisation : diagnostic de la mesure dans laquelle des performances d'I/O élevées peuvent contribuer à des performances d'exécution des tâches médiocres. Les nœuds de travail exécutant le DataNode démon doivent également effectuer des tâches de mappage et de réduction . La persistance de TotalLoad valeurs élevées au fil du temps peut indiquer que des E/S élevées peuvent contribue r à de mauvaises performances. Des pics occasionnels de cette valeur ne sont pas inhabituels et ne sont généralement pas le signe d'un problème. Unités : nombre

Métriques de capacité de cluster

Les métriques suivantes indiquent les capacités actuelles ou cibles d'un cluster. Ces métriques sont disponibles uniquement lorsque la mise à l'échelle gérée ou l'arrêt automatique sont activés.

Pour les clusters composés de parcs d'instances, les métriques de capacité de cluster sont mesurées en `Units`. Pour les clusters composés de groupes d'instances, les métriques de capacité de

cluster sont mesurées en `Nodes` ou en `VCPU` selon le type d'unité utilisé dans la politique de dimensionnement géré. Pour plus d'informations, consultez Utilisation de la mise à l'échelle gérée par EMR dans le Guide de gestion Amazon EMR.

Métrique	Description
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	Nombre total cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités : nombre
• `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 : nombre
• `CoreUnitsRequested` • `CoreNodesRequested` • `CoreVCPURequested`	Nombre cible d'unités/nœuds/vCPU CORE dans un cluster tel que déterminé par le dimensionnement géré. Unités : nombre
• `CoreUnitsRunning` • `CoreNodesRunning` • `CoreVCPURunning`	Nombre actuel d'unités/nœuds/vCPU CORE en cours d'exécution dans un cluster. Unités : nombre
• `TaskUnitsRequested`	Nombre cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré.

Métrique	Description
• `TaskNodesRequested` • `TaskVCPURequested`	Unités : nombre
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Nombre actuel d'unités/nœuds/vCPU TASK en cours d'exécution dans un cluster. Unités : nombre

Amazon EMR émet les statistiques suivantes avec une granularité d'une minute lorsque vous activez la terminaison automatique à l'aide d'une politique de terminaison automatique. Certaines mesures ne sont disponibles que pour les versions 6.4.0 et ultérieures d'Amazon EMR. Pour en savoir plus sur l'arrêt automatique, consultez Utilisation d'une politique de résiliation automatique.

Métrique	Description
`TotalNotebookKernels`	Nombre total de noyaux de bloc-notes en cours d'exécution et inactifs sur le cluster. Cette métrique n'est disponible que pour les versions 6.4.0 et ultérieures d'Amazon EMR.
`AutoTerminationIsClusterIdle`	Indique si le cluster est en cours d'utilisation. La valeur 0 indique que le cluster est activemen t utilisé par l'un des composants suivants : • Une application YARN • HDFS

Métrique	Description
	• Un bloc-notes • Une interface utilisateur intégrée au cluster, telle que le serveur d'historique Spark La valeur 1 indique que le cluster est inactif. Amazon EMR vérifie l'inactivité continue du cluster (`AutoTerminationIsClusterIdle` = 1). Lorsque le temps d'inactivité d'un cluster est égal à la valeur `IdleTimeout` de votre politique de terminaison automatique, Amazon EMR met fin au cluster.

Dimensions pour les métriques Amazon EMR

Les données Amazon EMR peuvent être filtrées à l'aide des dimensions du tableau ci-dessous.

Dimension	Description
JobFlowId	Le même que l'ID de cluster, qui correspond à l'identifiant unique d'un cluster sous la forme `j-XXXXXXXXXXXXX`. Trouvez cette valeur en cliquant sur le cluster dans la console Amazon EMR.

Surveillance des événements Amazon EMR avec CloudWatch

Amazon EMR surveille les événements et conserve les informations les concernant pendant sept jours maximum dans la console Amazon EMR. Amazon EMR enregistre les événements en cas de modification de l'état des clusters, des groupes d'instances, des flottes d'instances, des politiques de mise à l'échelle automatique ou des étapes. Les événements enregistrent la date et l'heure auxquelles l'événement s'est produit, des détails sur les éléments concernés et d'autres points de données critiques.

Le tableau suivant répertorie les événements Amazon EMR sur EKS en indiquant l'état ou la modification d'état rattachée à chaque événement, sa gravité, son type, son code ainsi que les messages correspondants. Amazon EMR représente les événements sous forme d'objets JSON et les envoie automatiquement vers un flux d'événements. L'objet JSON est important lorsque vous configurez des règles pour le traitement des événements à l'aide d' CloudWatch Events, car les règles cherchent à correspondre aux modèles de l'objet JSON. Pour plus d'informations, consultez les sections Événements et modèles d'événements et événements Amazon EMR dans le guide de l'utilisateur Amazon CloudWatch Events.

> **Note**
>
> Pour nous assurer de vous fournir les informations les plus pertinentes, nous affinons continuellement nos messages d'erreur. Pour cette raison, nous vous recommandons de ne pas analyser le texte des messages pour lancer les actions suivantes dans votre flux de travail.

Événements de démarrage de clusters

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`CREATING`	`WARN`	Mise en service de la flotte d'instances Amazon EMR	Mise en service EC2 : Capacité d'instance insuffisante	Nous ne sommes pas en mesure de créer votre cluster Amazon EMR `ClusterId (ClusterName)` . La flotte d'instances `InstanceFleetID` Amazon EC2 ne dispose pas d'une capacité Spot suffisant

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
				e pour le type d'instance `[Instance type1, Instancetype2]` ni d'une capacité à la demande suffisante pour le type d'instance `[Instance type3, Instancetype4]` dans la zone de disponibilité `[AvailabilityZone1, AvaliabilityZone2]` . Consultez la documentation ici pour plus d'informations sur la manière de réagir à cet événement.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`CREATING`	`WARN`	Mise en service du groupe d'instances Amazon EMR	Mise en service EC2 : Capacité d'instance insuffisante	Nous ne sommes pas en mesure de créer votre cluster Amazon EMR `ClusterId (ClusterName)` . Le groupe d'instances `InstancegroupID` Amazon EC2 ne dispose pas d'une capacité `[Spot or On-Demand]` suffisante pour le type d'instance `Instancetype` dans la zone de disponibilité `AvailabilityZone` . Consultez la documentation ici pour plus d'informations sur la manière de réagir à cet événement.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`STARTING`	`INFO`	Modification de l'état du cluster EMR	Aucun(e)	Le cluster Amazon EMR `ClusterId (ClusterName)` a été demandé à `Time` et est en cours de création.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`STARTING`	`INFO`	Modification de l'état du cluster EMR	Aucun(e)	**Note** S'applique uniquement aux clusters avec la configuration des parcs d'instances et plusieurs zones de disponibilité sélectionnées au sein de Amazon EC2. Le cluster Amazon EMR `ClusterId (ClusterName)` est créé dans la zone (`AvailabilityZoneID`), qui a été

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
				choisie parmi les options de zone de disponibilité spécifiées.
`STARTING`	`INFO`	Modification de l'état du cluster EMR	Aucun(e)	Le cluster Amazon EMR `ClusterId (ClusterName)` a commencé à exécuter des étapes à `Time`.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`WAITING`	`INFO`	Modification de l'état du cluster EMR	Aucun(e)	Le cluster Amazon EMR `ClusterId (ClusterName)` a été créé à `Time` et est prêt à être utilisé. - ou - Le cluster Amazon EMR `ClusterId (ClusterName)` a terminé d'exécuter toutes les étapes en attente à `Time`. **Note** Un cluster à l'état `WAITING` peut toujours traiter des tâches.

> **Note**
>
> Lors de la création ou de l'opération de redimensionnement du cluster, les événements rattachés au code d'événement `EC2 provisioning - Insufficient Instance Capacity` sont émis régulièrement lorsque votre cluster EMR rencontre une erreur de capacité insuffisante de la part d'Amazon EC2 pour votre flotte d'instances ou votre groupe d'instances. Pour plus d'informations sur comment réagir à ces événements, consultez Réponse aux événements liés à une capacité d'instance insuffisante du cluster Amazon EMR.

Événements de terminaison d'un cluster

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`TERMINATED`	La gravité dépend de la raison du changement d'état, comme illustré dans les exemples suivants : • **`CRITICAL`** si le cluster a terminé avec l'une des raisons de changement d'état suivantes : `INTERNAL_ERROR` ,	Modification de l'état du cluster EMR	Aucun(e)	Le cluster Amazon EMR `ClusterId (ClusterName)` a été terminé à `Time` pour le motif de `StateChangeReason: Code` .

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
	`VALIDATION_ERROR` , `INSTANCE_FAILURE` , `BOOTSTRAP_FAILURE` ou `STEP_FAILURE` . • **`INFO`** si le cluster a terminé avec l'une des raisons de changement d'état suivantes : `USER_REQUEST` ou `ALL_STEPS_COMPLETED` .			

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`TERMINATED_WITH_ERRORS`	`CRITICAL`	Modification de l'état du cluster EMR	Aucun(e)	Le cluster Amazon EMR `ClusterId (ClusterName)` a été terminé avec erreurs à `Time` pour le motif de `StateChangeReason: Code` .

Événements de modification de l'état de la flotte d'instances

> **Note**
>
> La configuration de flotte 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	Sévérité	Type d'événement	Code de l'événement	Message
De `PROVISIONING` à `WAITING`	`INFO`		Aucun(e)	La mise en service de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId`

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
				`(ClusterName)` est terminé. La mise en service a démarré à `Time` et a duré `Num` minutes. La flotte d'instances a maintenant une capacité à la demande de `Num` et une capacité Spot de `Num`. La capacité à la demande cible était de `Num` et la capacité Spot cible était de `Num`.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
De `WAITING` à `RESIZING`	`INFO`		Aucun(e)	Un redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` a débuté à `Time`. La flotte d'instances est en cours de redimensionnement depuis une capacité à la demande de `Num` vers une cible de `Num`, et depuis une capacité Spot de `Num` vers une cible de `Num`.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
De `RESIZING` à `WAITING`	`INFO`		Aucun(e)	L'opération de redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` est terminée. Le redimensionnement a démarré à `Time` et a duré `Num` minutes. La flotte d'instances a maintenant une capacité à la demande de `Num` et une capacité Spot de `Num`. La capacité à la demande cible était de `Num` et la capacité Spot cible était de `Num`.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
De `RESIZING` à `WAITING`	`INFO`		Aucun(e)	L'opération de redimensionnement pour la flotte d'instances `InstanceFleetID` du cluster Amazon EMR `ClusterId (ClusterName)` a dépassé le délai et s'est arrêtée. Le redimensionnement a démarré à `Time` et s'est arrêté après `Num` minutes. La flotte d'instances a maintenant une capacité à la demande de `Num` et une capacité Spot de `Num`. La capacité à la demande cible était de `Num` et la capacité Spot cible était de `Num`.

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
SUSPENDED	ERROR		Aucun(e)	La flotte d'instances InstanceFleetID du cluster Amazon EMR ClusterId (ClusterName) a été terminée à Time pour le motif suivant : ReasonDesc .
RESIZING	WARNING		Aucun(e)	L'opération de redimensionnement de la flotte d'instances InstanceFleetID dans le cluster Amazon EMR ClusterId (ClusterName) est bloquée pour le motif suivant : ReasonDesc .

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
`WAITING` ou `Running`	`INFO`		Aucun(e)	L'opération de redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` n'a pas pu être terminée, car Amazon EMR a ajouté de la capacité Spot dans la zone de disponibilité `AvailabilityZone`. Nous avons annulé votre demande de mise à disposition de capacité Spot supplémentaire. Pour connaître les actions recommandées, vérifiez Bonnes pratiques pour la flexibilité des

État ou changement d'état	Sévérité	Type d'événement	Code de l'événement	Message
				instances et des zones de disponibilité et réessayez.
`WAITING` ou `Running`	`INFO`		Aucun(e)	Une opération de redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` a été lancée par `Entity` à `Time`.

Événements de redimensionnement de la flotte d'instances

Type d'événement	Sévérité	Code de l'événement	Message
Redimensionnement de la flotte d'instances Amazon EMR	`ERROR`	Délai d'expiration pour la mise en service Spot	L'opération de redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` n'a pas pu être

Type d’événement	Sévérité	Code de l'événement	Message
			terminée lors de l'acquisition de capacité Spot dans la zone de disponibilité `AvailabilityZone` . Nous avons maintenant annulé votre demande et avons cessé d'essayer de fournir une capacité Spot supplémentaire. La flotte d'instances a fourni une capacité Spot de `num`. La capacité Spot cible était de `num`. Pour plus d'informations et les actions recommandées, consultez la page de documentation ici et réessayez.

Type d’événement	Sévérité	Code de l'événement	Message
Redimensionnement de la flotte d'instances Amazon EMR	ERROR	Délai d'expiration de la mise en service à la demande	L'opération de redimensionnement de la flotte d'instanc es `InstanceF leetID` dans le cluster Amazon EMR `ClusterId (Cluster Name)` n'a pas pu être terminée lors de l'acquisition de capacité à la demande dans la zone de disponibi lité `Availabil ityZone` . Nous avons maintenant annulé votre demande et avons cessé d'essayer de fournir une capacité à la demande supplémen taire. La flotte d'instances a fourni une capacité à la demande de `num`. La capacité à la demande cible était `num`. Pour plus d'informations et les actions recommand ées, consultez la page de documentation ici et réessayez.

Type d’événement	Sévérité	Code de l'événement	Message
Redimensionnement de la flotte d'instances Amazon EMR	`WARNING`	Mise en service EC2 : Capacité d'instance insuffisante	Nous ne sommes pas en mesure de terminer l'opération de redimensionnement de la flotte d'instances `InstanceFleetID` dans le cluster EMR `ClusterId (ClusterName)` , car Amazon EC2 ne dispose pas d'une capacité Spot suffisante pour les types d'instances `[Instancetype1, Instancetype2]` ni d'une capacité à la demande suffisante pour les types d'instances `[Instancetype3, Instancetype4]` dans la zone de disponibilité `[AvailabilityZone1]` . Jusqu'à présent, la flotte d'instances a mis en service une capacité à la demande de num ; la capacité à la demande cible était num. La capacité Spot

Type d’événement	Sévérité	Code de l'événement	Message
			allouée est num et la capacité Spot cible était num. Consultez la documentation ici pour plus d'informations sur la manière de réagir à cet événement.

Type d’événement	Sévérité	Code de l'événement	Message
Redimensionnement de la flotte d'instances Amazon EMR	`WARNING`	Délai d'expiration de la mise en service Spot : Redimensionnement continu	Nous sommes toujours en train de fournir de la capacité Spot pour l'opération de redimensionnement de la flotte d'instances qui a été lancée à `time`, par exemple, au niveau de l'ID de flotte `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` pour `[Instancetype1, Instancetype2]` dans la zone de disponibilité `AvailabilityZone`. Lors de la précédente opération de redimensionnement qui avait débuté à `time`, le délai d'expiration était dépassé. Amazon EMR a donc cessé de fournir de la capacité Spot après avoir ajouté `num` des instances `num`demandées à

Type d’événement	Sévérité	Code de l'événement	Message
			votre flotte d'instances. Pour plus d'informations, veuillez consulter la page de documentation ici.

Type d’événement	Sévérité	Code de l'événement	Message
Redimensionnement de la flotte d'instances Amazon EMR	`WARNING`	Délai d'expiration de la mise en service à la demande : Redimensionnement continu	Nous sommes toujours en train de fournir de la capacité à la demande pour l'opération de redimensionnement de la flotte d'instances qui a été lancée à `time`, par exemple, au niveau de l'ID de flotte `InstanceFleetID` dans le cluster Amazon EMR `ClusterId (ClusterName)` pour `[Instancetype1, Instancetype2]` dans la zone de disponibilité `AvailabilityZone`. Lors de la précédente opération de redimensionnement qui avait débuté à `time`, le délai d'expiration était dépassé. Amazon EMR a donc cessé de fournir de la capacité à la demande après avoir ajouté `num` des instances

Type d’événement	Sévérité	Code de l'événement	Message
			numdemandées à votre flotte d'instances. Pour plus d'informations, veuillez consulter la page de documentation ici.

> **Note**
>
> Les événements liés au délai de mise en service sont émis lorsqu'Amazon EMR arrête de fournir de la capacité Spot ou à la demande pour la flotte après l'expiration du délai. Pour plus d'informations sur comment réagir à ces événements, consultez Répondre aux événements d'expiration liés au redimensionnement de la flotte d'instances du cluster Amazon EMR.

Événements de groupe d'instances

Type d’événement	Sévérité	Code de l'événement	Message
De `RESIZING` à `Running`	`INFO`	Aucun(e)	L'opération de redimensionnement du groupe d'instances `InstanceGroupID` dans le cluster Amazon EMR `ClusterId (ClusterName)` est terminée. Le nombre d'instances est désormais de `Num`. Le redimensionnement a démarré à `Time` et a duré `Num`

Type d’événement	Sévérité	Code de l'événement	Message
			minutes avant de se terminer.
De `RUNNING` à `RESIZING`	`INFO`	Aucun(e)	Un redimensionnement du groupe d'instances `InstanceGroupID` dans le cluster Amazon EMR `ClusterId (ClusterName)` a débuté à `Time`. Le redimensionnement fait passer le nombre d'instances de `Num` à `Num`.
`SUSPENDED`	`ERROR`	Aucun(e)	Le groupe d'instances `InstanceGroupID` du cluster Amazon EMR `ClusterId (ClusterName)` a été terminé à `Time` pour le motif suivant : `ReasonDesc` .

Type d’événement	Sévérité	Code de l'événement	Message
RESIZING	WARNING	Aucun(e)	L'opération de redimensionnement du groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) est bloquée pour le motif suivant : ReasonDesc .

Type d’événement	Sévérité	Code de l'événement	Message
Redimensionnement du groupe d'instances Amazon EMR	`WARNING`	Mise en service EC2 : Capacité d'instance insuffisante	Nous ne sommes pas en mesure de terminer l'opérati on de redimensi onnement qui a débuté à `time` pour le groupe d'instances `InstanceGroupID` dans le cluster EMR `ClusterId (ClusterName)` , car Amazon EC2 ne dispose pas d'une capacité `Spot/On Demand` suffisante pour le type d'instanc e `[Instance type]` dans la zone de disponibi lité `[Availabi lityZone1]` . Jusqu'à présent, le nombre d'instances en cours d'exécution du groupe d'instanc es était de `num` et le nombre d'instanc es demandées était de `num`. Consultez la documentation ici pour plus d'informa tions sur la manière de réagir à cet événement.

Type d'événement	Sévérité	Code de l'événement	Message
De `RUNNING` à `RESIZING`	`INFO`	Aucun(e)	Un redimensionnement du groupe d'instances `InstanceGroupID` dans le cluster Amazon EMR `ClusterId (ClusterName)` a été lancé par `Entity` à `Time`.

> **Note**
>
> Avec la version 5.21.0 et ultérieures d'Amazon EMR, vous pouvez remplacer les configurations de cluster et de spécifier des classifications de configuration supplémentaires pour chaque groupe d'instances dans un cluster en cours d'exécution. Pour ce faire, utilisez la console Amazon EMR, le AWS Command Line Interface (AWS CLI) ou le AWS SDK. Pour plus d'informations, consultez 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	Sévérité	Message
`RUNNING`	`INFO`	Une reconfiguration du groupe d'instances `InstanceGroupID` dans le cluster Amazon EMR `ClusterId (ClusterName)` a été lancé par l'utilisateur à `Time`.

État ou changement d'état	Sévérité	Message
		La version de configuration demandée est Num.
De RECONFIGURING à Running	INFO	L'opération de reconfiguration du groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) est terminée. La reconfiguration a démarré à Time et a duré Num minutes. La version de configuration actuelle est Num.
De RUNNING à RECONFIGURING dans	INFO	Une reconfiguration du groupe d'instances InstanceGroupID dans le ClusterId (ClusterName) cluster Amazon EMR a débuté à Time. Il s'agit d'une configuration de la version Num vers la version Num.
RESIZING	INFO	L'opération de reconfiguration vers la version de configuration Num pour le groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) est temporairement bloquée à Time, car le groupe d'instances est à l'état State.

État ou changement d'état	Sévérité	Message
RECONFIGURING	INFO	L'opération de redimensionnement vers le nombre d'instances Num pour le groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) est temporairement bloquée à Time, car le groupe d'instances est à l'état State.
RECONFIGURING	WARNING	L'opération de reconfiguration du groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) a échoué à Time et mis Num minutes à échouer. La version de la configuration en échec est Num.
RECONFIGURING	INFO	Les configurations reviennent au numéro de version précédent Num pour le groupe d'instances InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterName) à Time. La nouvelle version de configuration est Num.

État ou changement d'état	Sévérité	Message
De RECONFIGURING à Running	INFO	Les configurations sont revenues avec succès au numéro de version précédent Num pour le groupe d'instanc es InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterN ame) à Time. La nouvelle version de configuration est Num.
De RECONFIGURING à SUSPENDED	CRITICAL	Impossible de revenir à la version réussie Num précédent e pour le groupe d'instanc es InstanceGroupID dans le cluster Amazon EMR ClusterId (ClusterN ame) à Time.

Événements de politique de mise à l'échelle automatique

État ou changement d'état	Sévérité	Message
PENDING	INFO	Une politique autoscaling a été ajoutée au groupe d'instanc es InstanceGroupID du cluster Amazon EMR ClusterId (Cluster Name) à Time. La politique est en attente d'attachement. - ou - La politique autoscaling pour le groupe d'instanc

État ou changement d'état	Sévérité	Message
		es `InstanceGroupID` du cluster Amazon EMR `ClusterId (Cluster Name)` a été mise à jour à `Time`. La politique est en attente d'attachement.
`ATTACHED`	`INFO`	La politique autoscaling pour le groupe d'instanc es `InstanceGroupID` du cluster Amazon EMR `ClusterId (Cluster Name)` a été rattachée à `Time`.
`DETACHED`	`INFO`	La politique autoscaling pour le groupe d'instanc es `InstanceGroupID` du cluster Amazon EMR `ClusterId (Cluster Name)` a été détachée à `Time`.

État ou changement d'état	Sévérité	Message
FAILED	ERROR	La politique autoscaling pour le groupe d'instanc es InstanceGroupID du cluster Amazon EMR ClusterId (Cluster Name) n'a pas pu être rattachée et a échoué à Time. - ou - La politique autoscaling pour le groupe d'instanc es InstanceGroupID du cluster Amazon EMR ClusterId (Cluster Name) n'a pas pu être détachée et a échoué à Time.

Événements d'étape

État ou changement d'état	Sévérité	Message
PENDING	INFO	L'étape StepID (StepName) a été ajoutée au cluster Amazon EMR ClusterId (ClusterName) à Time et est en attente d'exécution.
CANCEL_PENDING	WARN	L'étape StepID (StepName) du cluster Amazon EMR ClusterId (ClusterN ame) a été annulée à Time et est en attente d'annulation.

État ou changement d'état	Sévérité	Message
RUNNING	INFO	L'étape StepID (StepName) du cluster Amazon EMR ClusterId (ClusterName) a commencé à s'exécuter à Time.
COMPLETED	INFO	L'étape StepID (StepName) du cluster Amazon EMR ClusterId (ClusterName) s'est terminée à Time. L'étape a commencé à s'exécuter à Time et a duré Num minutes avant de se terminer.
CANCELLED	WARN	La demande d'annulation a réussi pour l'étape de cluster StepID (StepName) dans le cluster Amazon EMR ClusterId (ClusterName) à Time, et l'étape est maintenant annulée.
FAILED	ERROR	L'étape StepID (StepName) du cluster Amazon EMR ClusterId (ClusterName) a échoué à Time.

Événements de remplacement de nœuds défectueux

Type d'événement	Sévérité	Code de l'événement	Message
Remplacement d'un nœud défectueux d'Amazon EMR	INFO	Nœud principal défectueux détecté	Amazon EMR a identifié que l'instance principale du [instanceID (Instance Name)] cluster InstanceGroup/Fleet Amazon EMR est. clusterID (ClusterName) UNHEALTHY Amazon EMR tentera de récupérer ou de remplacer correctement l'instance. UNHEALTHY
Remplacement d'un nœud défectueux d'Amazon EMR	INFO	Nœud principal défectueux - remplacement désactivé	Amazon EMR a identifié que l'instance principale du [instanceID (Instance Name)] cluster InstanceGroup/Fleet Amazon

Type d’événement	Sévérité	Code de l'événement	Message	
			EMR est. `{clusterID}` `(ClusterName)` `UNHEALTHY` Activez le remplacement progressif des nœuds principaux défectueux de votre cluster pour permettre à Amazon EMR de remplacer `UNHEALTHY` les instances en cas d'impossibilité de les récupérer.	

Type d'événement	Sévérité	Code de l'événement	Message	
Remplacement d'un nœud défectueux d'Amazon EMR	WARN	Le nœud principal défectueux n'a pas été remplacé	Amazon EMR ne peut pas remplacer votre instance *UNHEALTHY* principale *[instanceID (InstanceName)]* *InstanceGroup/Fleet* dans le cluster *clusterID (ClusterName)* Amazon EMR pour une raison quelconque. **Note** La raison pour laquelle Amazon EMR ne peut pas remplacer votre nœud principal varie en fonction de votre	

Type d’événement	Sévérité	Code de l'événement	Message	
			scénario. Par exemple, l'une des raisons pour lesquelles Amazon EMR ne peut pas supprimer un nœud est qu'il ne resterait aucun nœud principal dans un cluster.	

Type d'événement	Sévérité	Code de l'événement	Message	
Remplacement d'un nœud défectueux d'Amazon EMR	INFO	Nœud principal défectueux restauré	Amazon EMR a récupéré vos instances UNHEALTHY principales [instanceID (Instance Name)] InstanceGroup/Fleet dans le cluster Amazon EMR clusterID (ClusterName)	

Pour plus d'informations sur le remplacement de nœuds défectueux, consultez la section Remplacement de nœuds défectueux.

Afficher des événements avec 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.

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 politique attachée à l'utilisateur.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Utiliser la console pour afficher des événements de tous les clusters d'une région

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Événements.

Utiliser la nouvelle console pour afficher les événements pour un cluster particulier

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis choisissez un cluster.
3. Pour afficher tous vos événements, sélectionnez l'onglet Événements sur la page de détails du cluster.

Old console

Utiliser l'ancienne console pour afficher des événements de tous les clusters d'une région

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Choisissez Events (Événements).

Utiliser l'ancienne console pour afficher les événements pour un cluster particulier

1. Ouvrez la console Amazon EMR à l'adressehttps://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.

Réagir aux CloudWatch événements

Cette section décrit les différentes manières de répondre aux événements exploitables émis par Amazon CloudWatch EMR sous forme de messages d'événement.

Rubriques

- Création de règles pour les événements Amazon EMR avec CloudWatch
- Configuration d'alarmes sur CloudWatch les métriques
- Réponse aux événements liés à une capacité d'instance insuffisante du cluster Amazon EMR
- Répondre aux événements d'expiration liés au redimensionnement de la flotte d'instances du cluster Amazon EMR

Création de règles pour les événements Amazon EMR avec CloudWatch

Amazon EMR envoie automatiquement les événements vers un flux d' CloudWatch événements. 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 plus d'informations sur les événements Amazon EMR, consultez les événements Amazon EMR dans le guide de l'utilisateur Amazon CloudWatch Events.

Pour plus d'informations sur la configuration des règles relatives aux CloudWatch événements, consultez la section Création d'une CloudWatch règle déclenchant un événement.

Configuration d'alarmes sur CloudWatch les métriques

Amazon EMR transmet les métriques à Amazon. CloudWatch En réponse, vous pouvez CloudWatch définir des alarmes sur vos métriques Amazon EMR. Par exemple, vous pouvez configurer une alarme pour vous CloudWatch envoyer un e-mail chaque fois que l'utilisation du HDFS dépasse 80 %. Pour obtenir des instructions détaillées, consultez la section Créer ou modifier une CloudWatch alarme dans le guide de CloudWatch l'utilisateur Amazon.

Réponse aux événements liés à une capacité d'instance insuffisante du cluster Amazon EMR

Présentation

Les clusters Amazon EMR renvoient le code d'événement `EC2 provisioning - Insufficient Instance Capacity` lorsque la zone de disponibilité sélectionnée n'a pas une capacité suffisante pour répondre à votre demande de démarrage ou de redimensionnement de cluster. L'événement est émis régulièrement, à la fois pour les groupes d'instances et les flottes d'instances, si Amazon EMR rencontre à plusieurs reprises des exceptions de capacité insuffisante et ne parvient pas à répondre à votre demande de mise en service pour une opération de démarrage ou de redimensionnement de cluster.

Cette page décrit la meilleure façon de répondre à ce type d'événement lorsqu'il se produit pour votre cluster EMR.

Réponse recommandée en cas d'insuffisance de capacité

En cas de capacité insuffisante, nous vous recommandons de réagir de l'une des manières suivantes :

- Attendez que la capacité soit rétablie. La capacité change fréquemment, ainsi, une exception de capacité insuffisante peut se résoudre elle-même. Le redimensionnement de vos clusters commencera ou se terminera dès que la capacité Amazon EC2 sera disponible.
- Vous pouvez également mettre fin à votre cluster, modifier les configurations de type d'instance et créer un nouveau cluster avec la demande de configuration de cluster mise à jour. Pour plus d'informations, consultez Bonnes pratiques pour la flexibilité des instances et des zones de disponibilité.

En cas de capacité insuffisante, vous pouvez également configurer des règles ou des réponses automatisées, comme décrit dans la section suivante.

Restauration automatique en cas d'insuffisance de capacité

Vous pouvez créer une automatisation en réponse aux événements Amazon EMR, tels que ceux comportant un code d'événement `EC2 provisioning - Insufficient Instance Capacity`. Par exemple, la AWS Lambda fonction suivante met fin à un cluster EMR avec un groupe d'instances qui utilise des instances à la demande, puis crée un nouveau cluster EMR avec un groupe d'instances contenant des types d'instances différents de ceux de la demande d'origine.

Les conditions suivantes déclenchent le processus automatisé :

- L'événement de capacité insuffisante est émis pour les nœuds principaux ou primaires depuis plus de 20 minutes.
- Le cluster n'est pas dans un état PRÊT ou EN ATTENTE. Pour de plus amples informations sur les états des clusters EMR, consultez Présentation du cycle de vie du cluster.

> **Note**
>
> Lorsque vous créez un processus automatisé pour une exception de capacité insuffisante, vous devez considérer que l'événement de capacité insuffisante est récupérable. La capacité change souvent et vos clusters reprendront leur redimensionnement ou commenceront à fonctionner dès que la capacité Amazon EC2 sera disponible.

Example Fonctionnalité pour répondre aux insuffisances de capacité

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE = "EMR Instance Group Provisioning"
INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE = (
    "EC2 provisioning - Insufficient Instance Capacity"
)
ALLOWED_INSTANCE_TYPES_TO_USE = [
    "m5.xlarge",
    "c5.xlarge",
    "m5.4xlarge",
    "m5.2xlarge",
    "t3.xlarge",
]
CLUSTER_START_ACCEPTABLE_STATES = ["WAITING", "RUNNING"]
CLUSTER_START_SLA = 20

CLIENT = boto3.client("emr", region_name="us-east-1")
```

```
# checks if the incoming event is 'EMR Instance Fleet Provisioning' with eventCode 'EC2
 provisioning - Insufficient Instance Capacity'
def is_insufficient_capacity_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceGroupType could be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]
    clusterCreationTime = describeClusterResponse["Cluster"]["Status"]["Timeline"][
        "CreationDateTime"
    ]
    clusterState = describeClusterResponse["Cluster"]["Status"]["State"]

    now = datetime.datetime.now()
    now = now.replace(tzinfo=timezone.utc)
    isClusterStartSlaBreached = clusterCreationTime < now - datetime.timedelta(
        minutes=CLUSTER_START_SLA
    )

    # Check if instance group receiving Insufficient capacity exception is CORE or
 PRIMARY (MASTER),
    # and it's been more than 20 minutes since cluster was created but the cluster
 state and the cluster state is not updated to RUNNING or WAITING
    if (
        (instanceGroupType == "CORE" or instanceGroupType == "MASTER")
        and isClusterStartSlaBreached
        and clusterState not in CLUSTER_START_ACCEPTABLE_STATES
    ):
        return True
    else:
        return False


# Choose item from the list except the exempt value
def choice_excluding(exempt):
```

```
    for i in ALLOWED_INSTANCE_TYPES_TO_USE:
        if i != exempt:
            return i


# Create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceGroupType cloud be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]

    # Following two lines assumes that the customer that created the cluster already
 knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # Select new instance types to include in the new createCluster request
    instanceTypeForMaster = (
        instanceTypesFromOriginalRequestMaster
        if instanceGroupType != "MASTER"
        else choice_excluding(instanceTypesFromOriginalRequestMaster)
    )
    instanceTypeForCore = (
        instanceTypesFromOriginalRequestCore
        if instanceGroupType != "CORE"
        else choice_excluding(instanceTypesFromOriginalRequestCore)
    )

    print("Starting to create cluster...")
    instances = {
        "InstanceGroups": [
            {
                "InstanceRole": "MASTER",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForMaster,
                "Market": "ON_DEMAND",
                "Name": "Master",
            },
            {
                "InstanceRole": "CORE",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForCore,
                "Market": "ON_DEMAND",
                "Name": "Core",
            },
```

```
        ]
    }
    response = CLIENT.run_job_flow(
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# Terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
    if is_insufficient_capacity_event(event):
        print(
            "Received insufficient capacity event for instanceGroup, clusterId: "
            + event["detail"]["clusterId"]
        )

        describeClusterResponse = describe_cluster(event)

        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
```

```
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
            )

    else:
        print("Received event is not insufficient capacity event, skipping")
```

Répondre aux événements d'expiration liés au redimensionnement de la flotte d'instances du cluster Amazon EMR

Présentation

Lors de l'exécution de l'opération de redimensionnement pour les clusters de flotte d'instances, les clusters Amazon EMR émettent des événements. Les événements liés au délai de mise en service sont émis lorsqu'Amazon EMR arrête de fournir de la capacité Spot ou à la demande pour la flotte après l'expiration du délai. La durée du délai d'expiration peut être configurée par l'utilisateur dans le cadre des spécifications de redimensionnement des flottes d'instances. Dans les scénarios de redimensionnement consécutifs pour la même flotte d'instances, Amazon EMR émet les événements `Spot provisioning timeout - continuing resize` ou `On-Demand provisioning timeout - continuing resize` lorsque le délai d'expiration est dépassé pour l'opération de redimensionnement en cours. Amazon EMR commence ensuite à fournir la capacité pour la prochaine opération de redimensionnement de la flotte.

Répondre aux événements liés au redimensionnement de la flotte d'instances

Lors d'un événement d'expiration de délai, nous vous recommandons de réagir de l'une des manières suivantes :

- Consultez les spécifications de redimensionnement et réessayez l'opération de redimensionnement. Comme la capacité change fréquemment, vos clusters seront redimensionnés avec succès dès que la capacité Amazon EC2 sera disponible. Nous recommandons aux clients de configurer des valeurs inférieures pour le délai d'expiration pour les tâches nécessitant des SLA plus stricts.
- Vous pouvez également :
  - Lancer un nouveau cluster avec des types d'instances diversifiés en fonction des meilleures pratiques pour les instances et de la flexibilité de la zone de disponibilité, ou
  - Lancement d'un cluster avec une capacité à la demande

- En ce qui concerne l'événement relatif au délai d'expiration de la mise en service et au redimensionnement continu, vous pouvez également attendre que les opérations de redimensionnement soient traitées. Amazon EMR continuera de traiter les opérations de redimensionnement déclenchées pour la flotte de manière séquentielle, en respectant les spécifications de redimensionnement configurées.

Vous pouvez également configurer des règles ou des réponses automatisées à cet événement, comme décrit dans la section suivante.

Restauration automatique après un événement d'expiration de la mise en service

Vous pouvez créer une automatisation en réponse aux événements Amazon EMR avec le code d'événement `Spot Provisioning timeout`. Par exemple, la fonction AWS Lambda suivante met fin à un cluster EMR avec une flotte d'instances qui utilise des instances Spot pour les nœuds de tâches, puis crée un nouveau cluster EMR avec une flotte d'instances contenant des types d'instances plus diversifiés que ceux de la demande initiale. Dans cet exemple, l'événement `Spot Provisioning timeout` émis pour les nœuds de tâches déclenchera l'exécution de la fonction Lambda.

Example Exemple de fonction pour répondre à un événement **`Spot Provisioning timeout`**

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE = "EMR Instance Fleet Resize"
SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE = (
    "Spot Provisioning timeout"
)

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Resize' with eventCode 'Spot
 provisioning timeout'
def is_spot_provisioning_timeout_event(event):
    if not event["detail"]:
        return False
```

```
    else:
        return (
            event["detail-type"] == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceFleetType could be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # Check if instance fleet receiving Spot provisioning timeout event is TASK
    if (instanceFleetType == "TASK"):
        return True
    else:
        return False


# create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceFleetType cloud be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # the following two lines assumes that the customer that created the cluster
 already knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # select new instance types to include in the new createCluster request
    instanceTypesForTask = [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge",
        "m5.8xlarge",
        "m5.12xlarge"
    ]

    print("Starting to create cluster...")
    instances = {
        "InstanceFleets": [
            {
                "InstanceFleetType":"MASTER",
```

```
                "TargetOnDemandCapacity":1,
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestMaster,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"CORE",
                "TargetOnDemandCapacity":1,
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestCore,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"TASK",
                "TargetOnDemandCapacity":0,
                "TargetSpotCapacity":100,
                "LaunchSpecifications":{},
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesForTask[0],
                        "WeightedCapacity":1,
                    },
                    {
                        'InstanceType': instanceTypesForTask[1],
                        "WeightedCapacity":2,
                    },
                    {
                        'InstanceType': instanceTypesForTask[2],
                        "WeightedCapacity":4,
                    },
                    {
                        'InstanceType': instanceTypesForTask[3],
                        "WeightedCapacity":8,
                    },
                    {
                        'InstanceType': instanceTypesForTask[4],
```

```
                        "WeightedCapacity":12,
                    }
                ],
                "ResizeSpecifications": {
                    "SpotResizeSpecification": {
                        "TimeoutDurationMinutes": 30
                    }
                }
            }
        ]
    }
    response = CLIENT.run_job_flow(
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
    if is_spot_provisioning_timeout_event(event):
        print(
            "Received spot provisioning timeout event for instanceFleet, clusterId: "
            + event["detail"]["clusterId"]
        )

        describeClusterResponse = describe_cluster(event)
```

```
        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
            )

    else:
        print("Received event is not spot provisioning timeout event, skipping")
```

Affichage des métriques d'application d'un cluster avec Ganglia

Ganglia est disponible avec les versions d'Amazon EMR entre 4.2 et 6.15. 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 Guide de version Amazon EMR.

Enregistrement des appels d'API Amazon EMR AWS CloudTrail

Amazon EMR est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans Amazon EMR. CloudTrail capture tous les appels d'API pour Amazon EMR sous forme d'événements. Ces captures incluent les appels de la console Amazon EMR et les appels de code vers les opérations d'API Amazon EMR. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon EMR. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon EMR, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

Informations Amazon EMR dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité a lieu dans Amazon EMR, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements relatifs à Amazon EMR, créez un historique. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les actions Amazon EMR sont enregistrées CloudTrail et documentées dans le Amazon EMR API Reference. Par exemple, les appels au `RunJobFlow` `ListCluster` et les `DescribeCluster` actions génèrent des entrées dans les fichiers CloudTrail journaux.

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

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Dans le cas où un processus, plutôt qu'un utilisateur, crée un cluster, vous pouvez utiliser l'identifiant `principalId` pour déterminer l'utilisateur rattaché à la création du cluster. Pour plus d'informations, consultez l'élément CloudTrail UserIdentity.

Exemple : entrées du fichier journal Amazon EMR

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant 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
   ]
}
```

Utiliser la mise à l'échelle des clusters

Vous pouvez ajuster le nombre d'instances Amazon EC2 disponibles dans un cluster Amazon EMR en fonction de charges de travail aux exigences diverses, soit automatiquement soit manuellement. Pour utiliser le dimensionnement automatique, vous avez deux options. Vous pouvez activer la mise à l'échelle gérée par Amazon EMR ou créer une politique de mise à l'échelle automatique personnalisé. Le tableau suivant décrit les différences entre les deux options.

	Mise à l'échelle gérée par Amazon EMR	Dimensionnement automatiq ue personnalisé
Stratégies et règles de dimensionnement	Aucune politique n'est requise. Amazon EMR gère l'activité de mise à l'échelle automatiq ue en évaluant en permanenc e les métriques de cluster et en prenant des décisions optimisées en matière de mise à l'échelle.	Vous devez définir et gérer les politiques et les règles de mise à l'échelle automatiq ue, telles que les conditions spécifiques qui déclenchent les activités de mise à l'échelle , les périodes d'évaluation, les temps de stabilisation, etc.
Versions Amazon EMR prises en charge	Amazon EMR versions 5.30.0 et ultérieures (sauf Amazon EMR version 6.0.0)	Amazon EMR versions 4.0.0 et ultérieures
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 dimension nement sont configurées pour l'ensemble du cluster.	Les limites de dimension nement 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 Amazon EMR de prendre des décisions de mise à l'échelle 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. La mise à l'échelle gérée par Amazon EMR ne prend pas en charge les applications	Vous pouvez choisir les applications prises en charge lors de la définition des règles de dimensionnement automatique.

	Mise à l'échelle gérée par Amazon EMR	Dimensionnement automatiq ue personnalisé
	qui ne sont pas basées sur YARN, telles que Presto ou HBase.	

Considérations

- Un cluster Amazon EMR se compose toujours d'un ou de trois nœuds primaires. Après la configuration initiale du cluster, vous ne pouvez mettre à l'échelle que les nœuds principaux et les nœuds de tâches. Vous ne pouvez pas mettre à l'échelle le nombre de nœuds primaires du cluster.
- Pour les groupes d'instances, les opérations de reconfiguration et de redimensionnement se produisent consécutivement et non simultanément. Si vous lancez une reconfiguration alors qu'un groupe d'instances est en cours de redimensionnement, la reconfiguration commence une fois que le groupe d'instances a terminé le redimensionnement en cours, et inversement si vous lancez une opération de redimensionnement alors qu'un groupe d'instance est en cours de reconfiguration.

Utiliser la mise à l'échelle gérée dans Amazon EMR

> **Important**
>
> Nous vous recommandons vivement d'utiliser la dernière version d'Amazon EMR (Amazon EMR 7.1.0) pour un dimensionnement géré. Dans certaines versions antérieures, vous pourriez rencontrer des défaillances intermittentes des applications ou des retards dans la mise à l'échelle. Amazon EMR a résolu ce problème dans les versions 5.30.2, 5.31.1, 5.32.1, 5.33.1 et les versions 5.x ultérieures, ainsi que dans les versions 6.1.1, 6.2.1 et 6.3.1 et les versions 6.x ultérieures. Pour de plus amples informations sur les régions et les zones de disponibilité, consultez Disponibilité de la mise à l'échelle gérée.

Présentation

Avec les versions 5.30.0 et ultérieures d'Amazon EMR (à l'exception d'Amazon EMR 6.0.0), vous pouvez activer la mise à l'échelle gérée par Amazon EMR. La mise à l'échelle gérée vous permet d'augmenter ou diminuer automatiquement le nombre d'instances ou d'unités dans votre cluster

en fonction de la charge de travail. Amazon 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. La mise à l'échelle automatique est disponible pour les clusters composés de groupes d'instances ou de flottes d'instances.

Disponibilité de la mise à l'échelle gérée

- Dans ce qui suit Régions AWS, le dimensionnement géré par Amazon EMR est disponible avec Amazon EMR 6.14.0 et versions ultérieures :
  - Asie-Pacifique (Hyderabad) (ap-south-2)
  - Asie-Pacifique (Jakarta) (ap-southeast-3)
  - Europe (Espagne) (eu-south-2)
- Dans ce qui suit Régions AWS, le dimensionnement géré par Amazon EMR est disponible avec Amazon EMR 5.30.0, 6.1.0 et versions ultérieures :
  - USA Est (Virginie du Nord) (us-east-1)
  - USA Est (Ohio) (us-east-2)
  - USA Ouest (Oregon) (us-west-2)
  - US Ouest (N. California) (us-west-1)
  - Afrique (Le Cap) (af-south-1)
  - Asie-Pacifique (Hong Kong) (ap-east-1)
  - Asie-Pacifique (Mumbai) (ap-south-1)
  - Asie-Pacifique (Séoul) (ap-northeast-2)
  - Asie-Pacifique (Singapour) (ap-southeast-1)
  - Asie-Pacifique (Sydney) (ap-southeast-2)
  - Asie-Pacifique (Tokyo) (ap-northeast-1)
  - Canada (Centre) (ca-central-1)
  - Amérique du Sud (São Paulo) (sa-east-1)
  - Europe (Francfort) (eu-central-1)
  - Europe (Irlande) (eu-west-1)
  - Europe (Londres) (eu-west-2)
  - Europe (Milan) (eu-south-1)
  - Europe (Paris) (eu-west-3)
  - Europe (Stockholm) (eu-north-1)

- Chine (Beijing) cn-north-1
- Chine (Ningxia) cn-northwest-1
- AWS GovCloud (Etats-Unis-Est) (us-gov-east-1)
- AWS GovCloud (US-Ouest) (us-gov-west-1)

- La mise à l'échelle gérée par Amazon EMR fonctionne uniquement avec des applications YARN, telles que Spark, Hadoop, Hive, Flink. Elle ne prend pas en charge les applications non basées sur YARN, telles que Presto et HBase.

Paramètres de mise à l'échelle gérée

Vous devez configurer les paramètres suivants pour la mise à l'échelle gérée. La limite s'applique uniquement aux nœuds principaux et aux nœuds de tâches. Le nœud primaire ne peut pas être mis à l'échelle 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 centrales virtuelles (vCPU) pour les groupes d'instances. Elle est mesuré par des unités, par exemple des flottes d'instance.
- Maximum (`MaximumCapacityUnits`) : limite supérieure de la capacité EC2 autorisée dans un cluster. Elle est mesurée par le biais de cœurs ou d'instances d'unités centrales virtuelles (vCPU) pour les groupes d'instances. Elle est mesuré par des unités, par exemple des flottes d'instance.
- Limite à la demande (`MaximumOnDemandCapacityUnits`) (facultatif) : 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, une valeur par défaut de `MaximumCapacityUnits` sera utilisée.
  - Ce paramètre est utilisé pour diviser l'allocation de capacité entre les instances à la demande et les instances Spot. Par exemple, si vous définissez le paramètre minimum à 2 instances, le paramètre maximum à 100 instances, la limite à la demande à 10 instances, la mise à l'échelle gérée par Amazon EMR augmente jusqu'à 10 instances à la demande et alloue la capacité restante aux instances Spot. Pour plus d’informations, consultez Scénarios d'allocation de nœuds.
- Nombre maximum de nœuds principaux (`MaximumCoreCapacityUnits`) (Facultatif) : limite supérieure de la capacité EC2 autorisée pour le type de nœud principal dans un cluster. Si ce paramètre n'est pas spécifié, une valeur par défaut de `MaximumCapacityUnits` sera utilisée.
  - Ce paramètre est utilisé pour diviser l'allocation de capacité entre les nœuds de base et les nœuds de tâche. Par exemple, si vous définissez le paramètre minimum à 2 instances, le maximum à 100 instances, le nœud principal maximal à 17 instances, la mise à l'échelle gérée

par Amazon EMR augmente jusqu'à 17 nœuds principaux et alloue les 83 instances restantes aux nœuds de tâches. Pour plus d'informations, consultez Scénarios d'allocation de nœuds.

Pour plus d'informations sur les paramètres de mise à l'échelle gérée, consultez `ComputeLimits`.

Considérations relatives à la mise à l'échelle gérée par Amazon EMR

- Le dimensionnement géré est pris en charge dans les versions limitées Régions AWS et dans les versions d'Amazon EMR. Pour plus d'informations, consultez Disponibilité de la mise à l'échelle gérée.
- Vous devez configurer les paramètres requis pour la mise à l'échelle gérée par Amazon EMR. Pour plus d'informations, consultez Paramètres de mise à l'échelle gérée.
- Pour utiliser la mise à l'échelle gérée, le processus de collecte de mesures doit pouvoir se connecter au point de terminaison d'API public pour une mise à l'échelle gérée dans la passerelle d'API. Si vous utilisez un nom DNS privé avec Amazon Virtual Private Cloud, le dimensionnement géré ne fonctionnera pas correctement. Pour garantir le bon fonctionnement de la mise à l'échelle gérée, nous vous recommandons de prendre l'une des mesures suivantes :
  - Supprimez le point de terminaison d'un VPC de l'interface de passerelle d'API de votre Amazon VPC.
  - Suivez les instructions de la section Pourquoi est-ce que je reçois une erreur d'accès interdit HTTP 403 lors de la connexion à mes API passerelles depuis un VPC ? pour désactiver le paramètre de nom DNS privé.
  - Lancez votre cluster dans un sous-réseau privé à la place. Pour plus d'informations, consultez la rubrique sur Sous-réseaux privés.
- Si vos tâches YARN sont ralenties par intermittence pendant la réduction et que les journaux du gestionnaire de ressources YARN indiquent que la plupart de vos nœuds ont été placés sur liste noire pendant cette période, vous pouvez ajuster le seuil de délai de mise hors service.

  Réduisez le `spark.blacklist.decommissioning.timeout` d'une heure à une minute pour que le nœud soit disponible et que d'autres conteneurs en attente puissent poursuivre le traitement des tâches.

  Vous devez également définir `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` sur une valeur plus élevée afin de vous assurer qu'Amazon EMR ne force pas la fermeture du nœud alors que la plus longue « tâche Spark » est toujours en cours d'exécution sur le nœud. La valeur par défaut actuelle est de 60 minutes, ce qui signifie que YARN

force la fermeture du conteneur au bout de 60 minutes une fois que le nœud entre en état de mise hors service.

L'exemple de ligne de journal du gestionnaire de ressources YARN suivant montre les nœuds ajoutés à l'état de mise hors service :

```
2021-10-20 15:55:26,994 INFO
 org.apache.hadoop.YARN.server.resourcemanager.DefaultAMSProcessor
 (IPC Server handler 37 on default port 8030): blacklist are updated in
 Scheduler.blacklistAdditions: [ip-10-10-27-207.us-west-2.compute.internal,
 ip-10-10-29-216.us-west-2.compute.internal, ip-10-10-31-13.us-
west-2.compute.internal, ... , ip-10-10-30-77.us-west-2.compute.internal],
 blacklistRemovals: []
```

Découvrez plus de détails sur la manière dont Amazon EMR s'intègre à la liste noire YARN lors de la mise hors service des nœuds, sur les cas où des nœuds Amazon EMR peuvent être répertoriés sur la liste de refus et sur la configuration du comportement de mise hors service des nœuds Spark.

- La surutilisation des volumes EBS peut entraîner des problèmes de mise à l'échelle gérée. Nous vous recommandons de maintenir le volume EBS en dessous de 90 % d'utilisation. Pour plus d'informations, consultez Stockage d'instances.
- Les CloudWatch métriques Amazon sont essentielles au bon fonctionnement de la mise à l'échelle gérée par Amazon EMR. Nous vous recommandons de suivre de près les CloudWatch métriques Amazon pour vous assurer que les données ne sont pas manquantes. Pour plus d'informations sur la façon dont vous pouvez configurer les CloudWatch alarmes afin de détecter les métriques manquantes, consultez Utiliser les CloudWatch alarmes Amazon.
- Les opérations de mise à l'échelle gérées sur des clusters 5.30.0 et 5.30.1 sans Presto installé peuvent provoquer des défaillances d'applications ou empêcher le maintien d'un groupe d'instances ou d'une flotte d'instances uniforme dans l'état `ARRESTED`, en particulier lorsqu'une opération de réduction est rapidement suivie d'une opération d'augementation.

  Pour contourner le problème, choisissez Presto comme application à installer lorsque vous créez un cluster avec les versions 5.30.0 et 5.30.1 d'Amazon EMR, même si votre travail ne nécessite pas Presto.
- Lorsque vous définissez le nombre maximal de nœuds principaux et la limite à la demande pour la mise à l'échelle gérée par Amazon EMR, tenez compte des différences entre les groupes d'instances et les flottes d'instances. Chaque groupe d'instances se compose du même type

d'instance et de la même option d'achat d'instances : À la demande ou Spot. Pour chaque flotte d'instances, vous pouvez indiquer jusqu'à cinq types d'instance, lesquels peuvent être mis en service en tant qu'instances À la demande ou en tant qu'instances Spot. Pour plus d'informations, consultez Créer un cluster avec des flottes d'instances ou des groupes d'instances uniformes, Options des flottes d'instances et Scénarios d'allocation de nœuds.

- Avec Amazon EMR 5.30.0 et versions ultérieures, si vous supprimez la règle Autoriser tous les accès sortants par défaut sur 0.0.0.0/ pour le groupe de sécurité principal, vous devez ajouter une règle qui autorise la connectivité TCP sortante à votre groupe de sécurité pour l'accès au service sur le port 9443. Votre groupe de sécurité pour l'accès aux services doit également autoriser le trafic TCP entrant sur le port 9443 à partir du groupe de sécurité principal. Pour plus d'informations sur la configuration des groupes de sécurité, consultez Groupe de sécurité géré par Amazon EMR pour l'instance principale (sous-réseaux privés).
- La mise à l'échelle gérée ne prend pas en charge la fonctionnalité d'étiquettes de nœuds YARN. Évitez d'utiliser des étiquettes de nœuds sur les clusters dotés d'une mise à l'échelle gérée. Par exemple, n'autorisez pas les exécuteurs à s'exécuter uniquement sur les nœuds de tâches. Lorsque vous utilisez des étiquettes de nœuds dans vos clusters Amazon EMR, il se peut que vous constatiez que votre cluster ne s'agrandit pas, ce qui peut entraîner un ralentissement de votre application.
- Vous pouvez l'utiliser AWS CloudFormation pour configurer le dimensionnement géré par Amazon EMR. Pour plus d'informations, consultez AWS::EMR::Clusterle guide de AWS CloudFormation l'utilisateur.

Historique des fonctionnalités

Ce tableau répertorie les mises à jour apportées à la fonctionnalité de mise à l'échelle gérée par Amazon EMR.

Date de publication	Capacité	Versions Amazon EMR
31 mars 2024	Le dimensionnement géré est disponible dans la région `ap-south-2` Asie-Pacifique (Hyderabad).	6.14.0 et ultérieures

Date de publication	Capacité	Versions Amazon EMR
13 février 2024	Le dimensionnement géré est disponible dans la région `eu-south-2` Europe (Espagne).	6.14.0 et ultérieures
10 octobre 2023	La mise à l'échelle gérée est disponible dans la Région Asie-Pacifique (Jakarta) `ap-southeast-3` .	6.14.0 et ultérieures
28 juillet 2023	Mise à l'échelle gérée améliorée pour passer à un autre groupe d'instances de tâches lors de l'augmentation quand Amazon EMR connaît un retard dans l'augmentation avec le groupe d'instances actuel.	5.34.0 et ultérieures, 6.4.0 et ultérieures
16 juin 2023	Mise à l'échelle gérée améliorée pour prendre en compte les nœuds exécutant le noeud principal de l'application master afin que ces nœuds ne soient pas réduits. Pour plus d'informations, consultez Comprendre la stratégie et les scénarios d'allocation des nœuds.	5.34.0 et ultérieures, 6.4.0 et ultérieures

Date de publication	Capacité	Versions Amazon EMR
21 mars 2022	Ajout de la fonction de reconnaissance des données de réorganisation Spark utilisée lors de la réduction de la taille des clusters. Pour les clusters Amazon EMR dotés d'Apache Spark et de la fonctionnalité de mise à l'échelle gérée activée, Amazon EMR surveille en permanence les exécuteurs Spark et les emplacements de données de réorganisation intermédiaires. À l'aide de ces informations, Amazon EMR réduit uniquement les instances sous-utilisées qui ne contiennent pas de données de réorganisation utilisées activement. Cela évite de recalculer les données de réorganisation perdues, ce qui contribue à réduire les coûts et à améliorer les performances au travail. Pour plus d'informations, consultez le Guide de programmation de Spark.	5.34.0 et ultérieures, 6.4.0 et ultérieures

Configuration de la mise à l'échelle gérée pour Amazon EMR

Les sections suivantes expliquent comment lancer un cluster EMR qui utilise le dimensionnement géré avec le AWS Management Console AWS SDK for Java, le ou le. AWS Command Line Interface

Rubriques

- Utilisez le AWS Management Console pour configurer le dimensionnement géré
- Utilisez le AWS CLI pour configurer le dimensionnement géré
- AWS SDK for Java À utiliser pour configurer le dimensionnement géré

Utilisez le AWS Management Console pour configurer le dimensionnement géré

Vous pouvez utiliser la console Amazon EMR pour configurer la mise à l'échelle gérée lorsque vous créez un cluster ou pour modifier une politique de mise à l'échelle gérée pour un cluster en cours d'exécution.

New console

Configurer la mise à l'échelle gérée lorsque vous créez un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Choisissez une version Amazon EMR emr-5.30.0 ou ultérieure, à l'exception de la version emr-6.0.0.
4. Sous l'option de mise à l'échelle et de mise en service du cluster, choisissez Utiliser la mise à l'échelle gérée par EMR. Spécifiez le nombre minimum et maximum d'instances, le nombre maximum d'instances de nœud principal et le nombre maximum d'instances à la demande.
5. Choisissez toutes les autres options qui s'appliquent à votre cluster.
6. Pour lancer cluster, choisissez Créer un cluster.

Configurer la mise à l'échelle gérée sur un cluster existant avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour.
3. Dans l'onglet Instances de la page de détails du cluster, recherchez la section Paramètres du groupe d'instances. Sélectionnez Modifier la mise à l'échelle du cluster pour spécifier de nouvelles valeurs pour le nombre minimum et maximum d'instances et la limite à la demande.

Old console

Lorsque vous créez un cluster dans l'ancienne console, vous pouvez configurer la mise à l'échelle gérée à l'aide d'options rapides ou d'options de configuration avancée du cluster. Vous pouvez également créer ou modifier une politique de mise à l'échelle gérée pour un cluster en cours d'exécution en modifiant les paramètres de mise à l'échelle gérée sur la page Récapitulatif ou Matériel.

Utiliser les options rapides afin de configurer la mise à l'échelle gérée lorsque vous créez un cluster avec l'ancienne console

1. Ouvrez la console Amazon EMR, choisissez Créer un cluster et ouvrez Créer un cluster : options rapides.
2. Dans la section Configuration matérielle à côté de l'option de mise à l'échelle et de mise en service du cluster, cochez la case pour activer la mise à l'échelle des nœuds de cluster en fonction de la charge de travail.
3. Sous Unités principales et de tâche, indiquez les nombres minimum et maximum d'instances principales et de tâche.

Utiliser l'option avancée pour configurer la mise à l'échelle gérée lorsque vous créez un cluster avec l'ancienne console.

1. Dans la console Amazon EMR, sélectionnez Créer un cluster, sélectionnez Accéder aux options avancées, choisissez les options pour Étape 1 : Logiciels et étapes, puis accédez à Étape 2 : Configuration du matériel.
2. Dans la section Cluster composition (Composition du cluster), sélectionnez Parcs d'instances ou Groupes d'instances uniformes.
3. Sous l'option de mise à l'échelle et mise en service du cluster, sélectionnez Activer la mise à l'échelle gérée du cluster. Sélectionnez ensuite Utiliser la mise à l'échelle gérée par EMR. Sous Unités principales et unités de tâches, spécifiez le nombre minimum et maximum d'instances ou d'unités de la flotte d'instances, la limite à la demande et le nombre maximal de nœuds principaux.

   Pour les clusters composés de groupes d'instances, vous pouvez également choisir Create a custom automatic scaling policy (Créer une politique de dimensionnement automatique personnalisée) si vous souhaitez définir des politiques de dimensionnement automatique personnalisées pour chaque groupe d'instances. Pour plus d’informations, consultez

Utilisation de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.

Modifier la mise à l'échelle gérée sur un cluster existant avec la nouvelle console

1. Ouvrez la console Amazon EMR, sélectionnez votre cluster dans la liste des clusters, puis choisissez l'onglet Matériel.
2. Dans la section Option de mise à l'échelle et de mise en service du cluster, sélectionnez Modifier pour la mise à l'échelle gérée par Amazon EMR.
3. Dans la section Option de mise à l'échelle et de mise en service du cluster, spécifiez de nouvelles valeurs pour le nombre minimum et maximum d'instances et la limite À la demande.

Utilisez le AWS CLI pour configurer le dimensionnement géré

Vous pouvez utiliser des AWS CLI commandes pour Amazon EMR afin 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-7.1.0 \
 --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 politique gérée à l'aide de l'managed-scaling-policy option -- 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 le 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
```

AWS SDK for Java À utiliser pour configurer le 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 Amazon EMR client with 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 - Primary, 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 Amazon 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-7.1.0")          // Specifies the version label for
the Amazon EMR release; 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 Amazon EMR
role for EC2 instance profile, replace the default role with the custom Amazon 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 getCredentials(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() { }
}
```

Comprendre la stratégie et les scénarios d'allocation des nœuds

Cette section donne un aperçu de la stratégie d'allocation des nœuds et des scénarios de mise à l'échelle courants que vous pouvez utiliser avec la mise à l'échelle gérée par Amazon EMR.

Stratégie d'allocation de nœud

La mise à l'échelle gérée par Amazon EMR alloue les nœuds principaux et les nœuds de tâches en fonction des stratégies d'augmentation et de réduction d'échelle suivantes :

Stratégie d'augmentation

- La mise à l'échelle gérée par Amazon EMR ajoute d'abord de 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 d'augmentation souhaitée soit atteinte.
- Lorsque Amazon EMR subit un retard dans l'augmentation avec le groupe d'instances actuel, les clusters qui utilisent la mise à l'échelle gérée basculent automatiquement vers un autre groupe d'instances de tâches.

- Si le paramètre `MaximumCoreCapacityUnits` est défini, Amazon EMR adapte 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 paramètre `MaximumOnDemandCapacityUnits` est défini, Amazon EMR met le cluster à l'échelle 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 d'instances Spot.
- Si les paramètres `MaximumCoreCapacityUnits` et `MaximumOnDemandCapacityUnits` sont définis, Amazon EMR prend en compte les deux limites lors de la mise à l'échelle.

  Par exemple, si `MaximumCoreCapacityUnits` est inférieur à `MaximumOnDemandCapacityUnits`, Amazon EMR augmente d'abord les nœuds principaux jusqu'à ce que la limite de capacité principale soit atteinte. Pour la capacité restante, Amazon EMR utilise d'abord des instances à la demande pour mettre à l'échelle les nœuds de tâches jusqu'à ce que la limite de la demande soit atteinte, puis utilise des instances Spot pour les nœuds de tâches.

Stratégie de réduction

- Les versions 5.34.0 et ultérieures, ainsi que les versions 6.4.0 et ultérieures d'Amazon EMR, prennent en charge la mise à l'échelle gérée qui tient compte des données de réorganisation Spar (données que Spark redistribue entre les partitions pour effectuer des opérations spécifiques). Pour plus d'informations sur les opérations de réorganisation, consultez le Guide de programmation Spark. La mise à l'échelle gérée réduit uniquement les instances sous-utilisées et qui ne contiennent pas de données de réorganisation utilisées activement. Cette mise à l'échelle intelligente empêche la perte involontaire de données par transfert, évitant ainsi de devoir effectuer de nouvelles tentatives de travail et de recalculer des données intermédiaires.
- La mise à l'échelle gérée par Amazon EMR supprime d'abord les nœuds de tâches, puis les nœuds principaux jusqu'à ce que la capacité cible de réduction souhaitée soit atteinte. Le cluster ne s'adapte jamais en dessous des contraintes minimales définies dans la politique de mise à l'échelle gérée.
- Au sein de chaque type de nœud (nœuds principaux ou nœuds de tâches), la mise à l'échelle gérée par Amazon EMR supprime d'abord les instances Spot, puis les instances à la demande.
- Pour les clusters lancés avec Amazon EMR 5.x, versions 5.34.0 et ultérieures, et 6.x, versions 6.4.0 et ultérieures, la mise à l'échelle gérée par Amazon EMR ne réduit pas la taille des nœuds sur lesquels `ApplicationMaster` pour Apache Spark est exécuté. Cela permet de minimiser les échecs et les nouvelles tentatives, ce qui contribue à améliorer les performances de tâche et à réduire les coûts. Pour vérifier quels nœuds de votre cluster exécutent

`ApplicationMaster`, rendez-vous sur le serveur d'historique Spark et filtrez le pilote sous l'onglet Exécuteurs de l'ID de votre application Spark.

Si le cluster n'est pas chargé, Amazon EMR annule l'ajout de nouvelles instances issues d'une évaluation précédente et effectue des opérations de réduction. Si le cluster est soumis à une charge importante, Amazon EMR annule la suppression des instances et effectue des opérations d'augmentation.

Considérations concernant l'allocation de nœuds

Nous vous recommandons d'utiliser l'option d'achat à la demande pour les nœuds principaux afin d'éviter toute perte de données HDFS en cas de réclamation Spot. Vous pouvez utiliser l'option d'achat Spot pour les nœuds de tâches afin de réduire les coûts et d'accélérer l'exécution des tâches lorsque davantage d'instances Spot sont ajoutées aux nœuds de tâches.

Scénarios d'allocation de nœuds

Vous pouvez créer différents scénarios de mise à l'échelle en fonction de vos besoins en configurant les paramètres maximum, minimum, limite à la demande et maximum du nœud principal dans différentes combinaisons.

Scénario 1 : mise à l'échelle des nœuds principaux uniquement

Pour mettre à l'échelle les nœuds principaux uniquement, les paramètres de mise à l'échelle gérée doivent répondre aux exigences suivantes :

- La limite à la demande est égale à la limite maximale.
- Le nœud principal maximal est égal à la limite maximale.

Lorsque la limite à la demande et les paramètres maximum du nœud principal ne sont pas spécifiés, les deux paramètres sont définis par défaut sur la limite maximale.

Les exemples suivants montrent le scénario de mise à l'échelle des nœuds principaux uniquement.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
Groupes d'instances	`UnitType` : Instances	

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
Principal : 1 à la demande De tâche : 1 à la demande et 1Spot	`MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 20	Mettre à l'échelle entre 1 et 20 instances ou unités de flotte d'instances sur les nœuds principaux en utilisant le type À la demande. Aucune mise à l'échelle sur les nœuds de tâches.
Flottes d'instances Principal : 1 à la demande De tâche : 1 à la demande et 1Spot	UnitType: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 20	

Scénario 2 : mettre à l'échelle les nœuds de tâches uniquement

Pour mettre à l'échelle les nœuds de tâches uniquement, les paramètres de mise à l'échelle gérée doivent répondre aux exigences suivantes :

- Le nœud principal maximal doit être égal à la limite minimale.

Les exemples suivants montrent le scénario de mise à l'échelle des nœuds de tâches uniquement.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
Groupes d'instances	`UnitType` : Instances	

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
Principal : 2 à la demande De tâche : 1 Spot	`MinimumCapacityUnits` : 2 `MaximumCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 2	Maintenez la stabilité des nœuds principaux à 2 et mettez à l'échelle uniquement les nœuds de tâches entre 0 et 18 instances ou unités de flotte d'instances. La capacité entre les limites minimale et maximale est ajoutée aux nœuds de tâches uniquement.
Flottes d'instances Principal : 2 à la demande De tâche : 1 Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 2 `MaximumCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 2	

Scénario 3 : uniquement les instances à la demande dans le cluster

Pour disposer uniquement d'instances à la demande, votre cluster et les paramètres de mise à l'échelle gérée doivent répondre aux exigences suivantes :

- La limite à la demande est égale à la limite maximale.

  Lorsque la limite à la demande n'est pas spécifiée, la valeur du paramètre est par défaut la limite maximale. La valeur par défaut indique qu'Amazon EMR met à l'échelle uniquement les instances à la demande.

Si le nœud principal maximal est inférieur à la limite maximale, le paramètre du nœud principal maximal peut être utilisé pour répartir l'allocation de capacité entre le nœud principal et le nœud de tâche.

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 consistant à disposer d'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 Principal : 1 à la demande Tâche : 1 à la demande	`UnitType` : Instances `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 12	Mettre à l'échelle entre 1 et 12 instances ou unités de flotte d'instances sur les nœuds principaux en utilisant le type À la demande. Augmentez la capacité restante en utilisant À la demande sur les nœuds de tâches. Aucune mise à l'échelle avec les instances Spot.
Flottes d'instances Principal : 1 à la demande Tâche : 1 à la demande	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 12	

Scénario 4 : uniquement les instances Spot du cluster

Pour disposer uniquement d'instances Spot, les paramètres de mise à l'échelle gérée doivent répondre aux exigences suivantes :

- La limite à la demande est fixée à 0.

Si le nœud principal maximal est inférieur à la limite maximale, le paramètre du nœud principal maximal peut être utilisé pour répartir l'allocation de capacité entre le nœud principal et le nœud de tâche.

Pour activer ce scénario dans un cluster composé de groupes d'instances, le groupe d'instances principal doit utiliser l'option d'achat Spot lors de la configuration initiale. S'il n'y a aucune instance Spot dans le groupe d'instances de tâches, la mise à l'échelle gérée par Amazon EMR crée un groupe de tâches utilisant des instances Spot en cas de besoin.

Les exemples suivants illustrent le scénario consistant à disposer d'instances Spot dans l'ensemble du cluster.

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
Groupes d'instances Principal : 1 Spot De tâche : 1 Spot	`UnitType` : Instances `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 0	Mettre à l'échelle entre 1 et 20 instances ou unités de flotte d'instances sur les nœuds principaux en utilisant Spot. Aucune mise à l'échelle avec le type À la demande.
Flottes d'instances Principal : 1 Spot De tâche : 1 Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 0	

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 mettre à l'échelle les instances à la demande sur les nœuds principaux et les instances Spot sur les nœuds de tâches, les paramètres de mise à l'échelle gérée doivent répondre aux exigences suivantes :

- La limite à la demande doit être égale au nombre maximal de nœuds principaux.
- La limite à la demande et le nombre maximal de nœuds principaux doivent être inférieurs à la limite maximale.

Pour activer ce scénario dans un cluster composé de groupes d'instances, le groupe de noeud principal doit utiliser l'option d'achat à la demande.

Les exemples suivants illustrent le scénario de mise à l'échelle 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 Principal : 1 à la demande De tâche : 1 à la demande et 1Spot	`UnitType` : Instances `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 7 `MaximumCoreCapacityUnits` : 7	Augmentez à 6 unités à la demande sur le nœud principal, car il existe déjà une unité à la demande sur le nœud de tâche et la limite maximale pour la demande est de 7. Augmentez ensuite à 13 unités Spot
Flottes d'instances Principal : 1 à la demande De tâche : 1 à la demande et 1Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20	

État initial du cluster	Paramètres de dimensionnement	Comportement de mise à l'échelle.
	`MaximumOnDemandCapacityUnits` : 7 `MaximumCoreCapacityUnits` : 7	sur les nœuds de tâches.

Présentation des métriques de mise à l'échelle gérée

Amazon EMR publie des métriques haute résolution avec des données à une granularité d'une minute lorsque la mise à l'échelle gérée est activé pour un cluster. Vous pouvez consulter les événements relatifs au lancement et à la fin de chaque redimensionnement grâce au dimensionnement géré à l'aide de la console Amazon EMR ou de la console Amazon CloudWatch . CloudWatch les métriques sont essentielles au bon fonctionnement de la mise à l'échelle gérée par Amazon EMR. Nous vous recommandons de suivre de près CloudWatch les indicateurs pour vous assurer que les données ne sont pas manquantes. Pour plus d'informations sur la façon dont vous pouvez configurer les CloudWatch alarmes afin de détecter les métriques manquantes, consultez Utiliser les CloudWatch alarmes Amazon. Pour plus d'informations sur l'utilisation CloudWatch des événements avec Amazon EMR, consultez Surveiller CloudWatch les événements.

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 de cluster sont mesurées en `Units`. Pour les clusters composés de groupes d'instances, les métriques de capacité de cluster sont mesurées en `Nodes` ou en `vCPU` selon le type d'unité utilisé dans la politique de dimensionnement géré.

Métrique	Description
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	Nombre total cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités : nombre

Métrique	Description
• `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 : nombre
• `CoreUnitsRequested` • `CoreNodesRequested` • `CoreVCPURequested`	Nombre cible d'unités/nœuds/vCPU CORE dans un cluster tel que déterminé par le dimensionnement géré. Unités : nombre
• `CoreUnitsRunning` • `CoreNodesRunning` • `CoreVCPURunning`	Nombre actuel d'unités/nœuds/vCPU CORE en cours d'exécution dans un cluster. Unités : nombre
• `TaskUnitsRequested` • `TaskNodesRequested` • `TaskVCPURequested`	Nombre cible d'unités/nœuds/vCPU dans un cluster tel que déterminé par le dimensionnement géré. Unités : nombre
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Nombre actuel d'unités/nœuds/vCPU TASK en cours d'exécution dans un cluster. Unités : nombre

Les métriques suivantes indiquent l'état d'utilisation du cluster et des applications. Ces mesures sont disponibles pour toutes les fonctionnalités d'Amazon EMR, 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 : surveiller la progression du cluster Unités : nombre
`AppsPending`	Nombre d'applications soumises à YARN qui se trouvent dans un état d'attente. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
`AppsRunning`	Nombre d'applications soumises à YARN qui sont en cours d'exécution. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
`ContainerAllocated`	Le nombre de conteneurs de ressources alloués par leResourceManager. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
`ContainerPending`	Nombre de conteneurs dans la file d'attente qui n'ont pas encore été alloués.

Métrique	Description
	Cas d'utilisation : surveiller la progression du cluster Unités : nombre
`ContainerPendingRatio`	Le rapport entre les conteneurs en attente et les conteneurs alloués (ContainerPendingRatio = ContainerPending / ContainerAllocated). Si ContainerAllocated = 0, alors ContainerPendingRatio =ContainerPending. La valeur de ContainerPendingRatio représente un nombre et non un pourcentage. Cette valeur est utile pour dimensionner les ressources de cluster en fonction du comportement d'attribution des conteneurs. Unités : nombre
`HDFSUtilization`	Pourcentage de stockage HDFS actuellement utilisé. Cas d'utilisation : analyser les performances du cluster Unités : pourcentage

Métrique	Description
`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 : surveiller les performances du cluster Unités : booléennes
`MemoryAvailableMB`	Quantité de mémoire disponible à allouer. Cas d'utilisation : surveiller la progression du cluster Unités : nombre
`MRActiveNodes`	Le nombre de nœuds exécutant actuellement MapReduce des tâches ou des tâches. Équivalent à la métrique YARN `mapred.resourcemanager.NoOfActiveNodes` . Cas d'utilisation : surveiller la progression du cluster Unités : nombre

Métrique	Description
YARNMemoryAvailablePercentage	Pourcentage de mémoire restante disponible pour YARN (YARN MemoryAvailablePercentage = MemoryAvailable MemoryTotal Mo/Mo). Cette valeur est utile pour dimensionner les ressources de cluster en fonction de l'utilisation de mémoire YARN. Unités : pourcentage

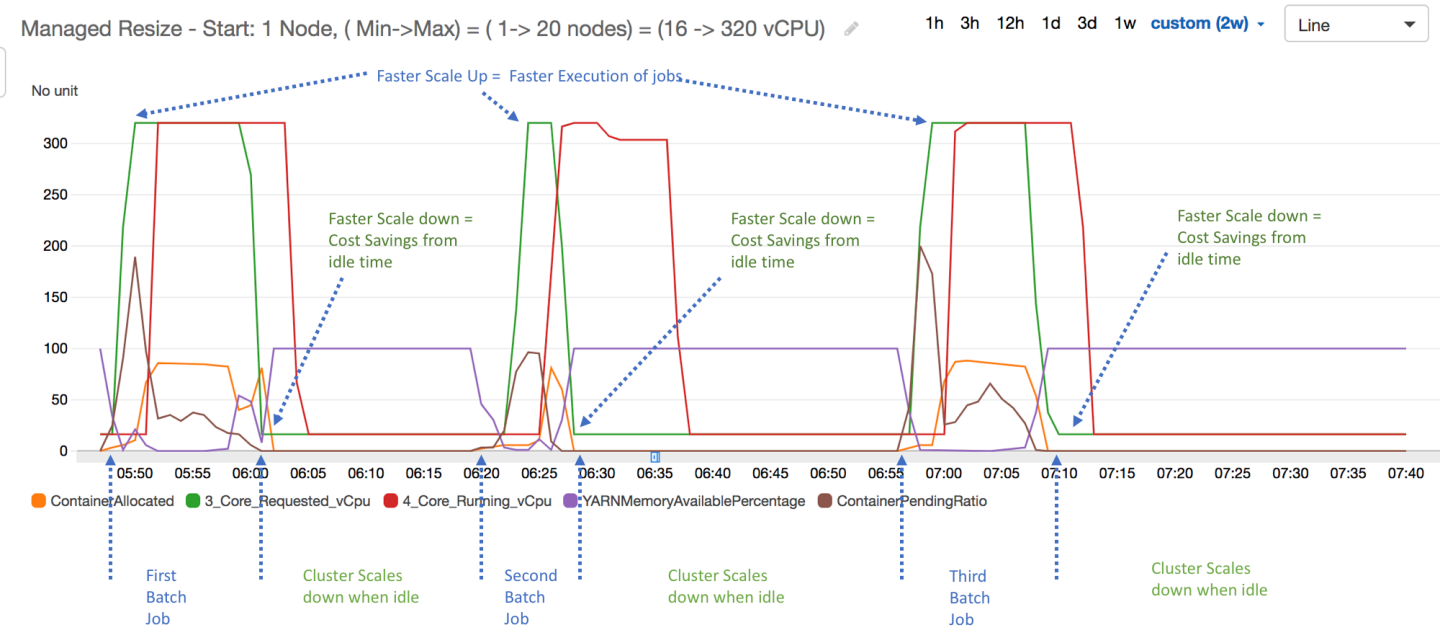
Représentation graphique des métriques de mise à l'échelle gérée

Vous pouvez représenter graphiquement les métriques pour visualiser les modèles de charge de travail de votre cluster et les décisions de mise à l'échelle correspondantes prises par la mise à l'échelle gérée par Amazon EMR, comme le montrent les étapes suivantes.

Pour représenter graphiquement les métriques de dimensionnement gérées dans la CloudWatch console

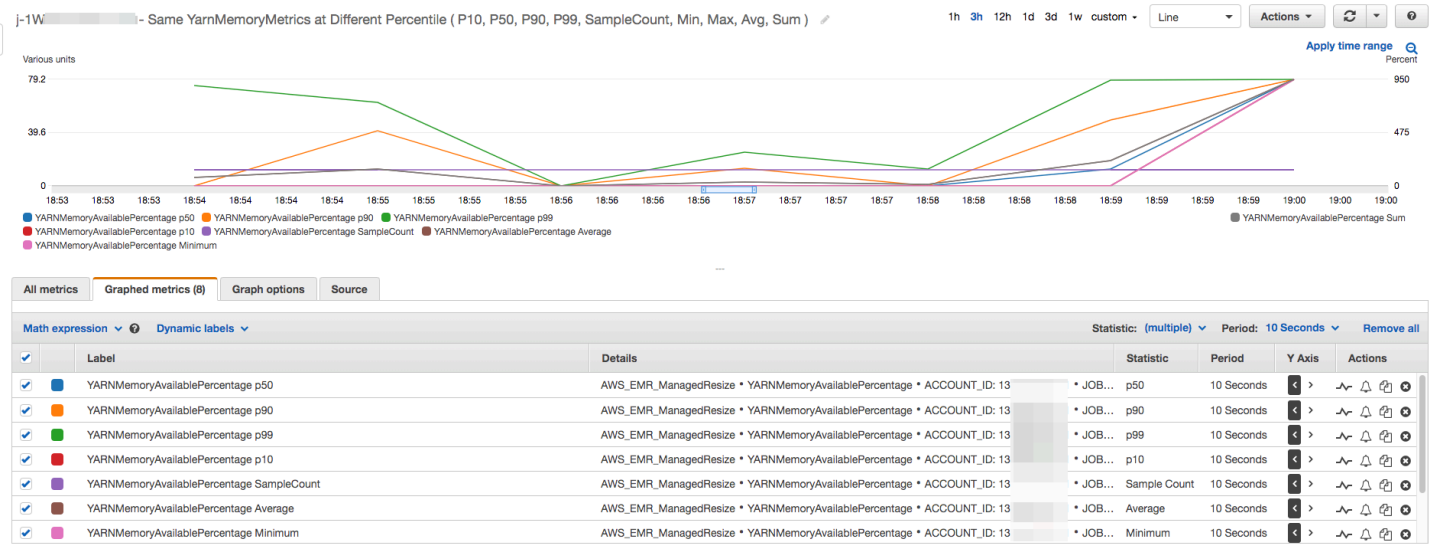
1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation, choisissez Amazon EMR. Vous pouvez rechercher par l'identifiant de cluster du cluster à surveiller.
3. Faites défiler jusqu'à la métrique que vous souhaitez représenter graphiquement. Ouvrez une métrique pour afficher le graphique.
4. Pour représenter graphiquement une ou plusieurs métriques, cochez la case en regard de chaque métrique.

L'exemple suivant illustre l'activité de mise à l'échelle gérée par Amazon EMR pour 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 rattachées 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 la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances

Le dimensionnement automatique avec une politique personnalisée dans les versions 4.0 et supérieures d'Amazon EMR vous permet de dimensionner et de dimensionner de manière programmatique les nœuds principaux et les nœuds de tâches en fonction d'une CloudWatch métrique et d'autres paramètres que vous spécifiez dans une politique 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.

> **Note**
>
> Pour utiliser la fonctionnalité de mise à l'échelle automatique avec une politique personnalisée dans Amazon EMR, vous devez définir `true` pour le paramètre `VisibleToAllUsers` lorsque vous créez un cluster. Pour plus d'informations, consultez la section SetVisibleToAllUtilisateurs.

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 d'un cluster, à l'exception du groupe d'instances principal, peut avoir sa propre politique de mise à l'échelle, qui consiste en des règles de mise à l'échelle (monter en puissance) et de mise à l'échelle (mise à l'échelle horizontale). Les règles de dimensionnement (augmentation et diminution) peuvent être configurées indépendamment, avec des paramètres différents pour chaque règle.

Vous pouvez configurer des politiques de dimensionnement à l'aide de l'API AWS Management Console AWS CLI, de, ou de l'API Amazon EMR. Lorsque vous utilisez l'API AWS CLI ou Amazon EMR, vous spécifiez la politique de dimensionnement au format JSON. En outre, lorsque vous utilisez l'API AWS CLI ou l'API Amazon EMR, vous pouvez spécifier des métriques personnalisées CloudWatch . Les métriques personnalisées ne sont pas disponibles pour la sélection avec la AWS Management Console. Lorsque vous créez une politique de mise à l'échelle avec la console, une politique par défaut adaptée à 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 du cluster EMR on-the-fly, vous devez toujours tenir compte des exigences de charge de travail de base et planifier les configurations de vos nœuds et de vos groupes d'instances. 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 la mise à l'échelle automatique

La mise à l'échelle automatique dans Amazon EMR nécessite un rôle IAM avec des autorisations pour ajouter et supprimer des instances lorsque les activités de mise à l'échelle sont déclenchées. 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 politique de dimensionnement pour la première fois avec le AWS Management Console, Amazon EMR crée le rôle par défaut et associe la politique gérée par défaut pour les autorisations,. `AmazonElasticMapReduceforAutoScalingRole`

Lorsque vous créez un cluster doté d'une politique de dimensionnement automatique avec le AWS CLI, vous devez d'abord vous assurer que le rôle IAM par défaut existe ou que vous disposez d'un rôle IAM personnalisé auquel est attachée une politique fournissant 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. Autrement, vous pouvez créer un rôle de dimensionnement automatique personnalisé, puis le spécifier lors de la création d'un cluster, par exemple `--auto-scaling-role` *`MyEMRAutoScalingRole`*. Si vous créez un rôle personnalisé de mise à l'échelle automatique pour Amazon EMR, nous vous recommandons de baser les politiques d'autorisation de votre rôle personnalisé sur la politique gérée. Pour plus d'informations, consultez Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..

Présentation des règles de dimensionnement automatique

Lorsqu'une règle de mise à l'échelle déclenche une activité de mise à l'échelle pour un groupe d'instances, des instances Amazon EC2 sont ajoutées au groupe d'instances conformément à 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 qui sont mises à l'échelle automatiquement, consultez Cycle de vie de l'autoscaling dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

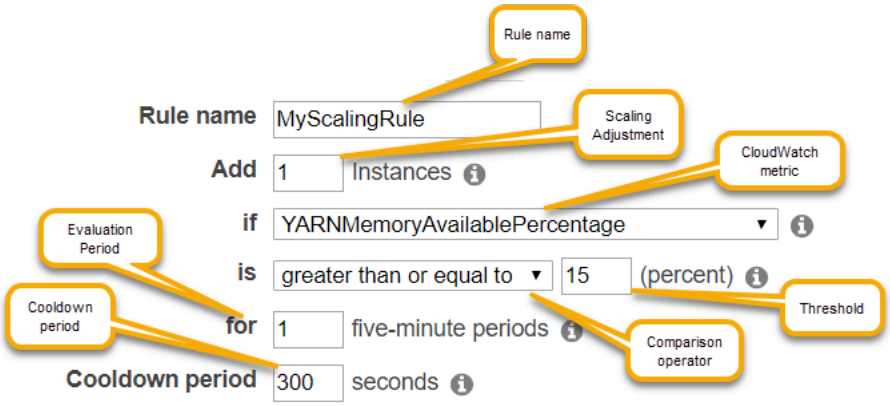
Vous pouvez configurer la façon dont un cluster met des instances Amazon EC2 hors service. Pour la facturation, vous pouvez choisir de mettre hors service à l'échéance horaire de l'instance Amazon EC2 ou lorsque la tâche est terminée. Ce paramètre s'applique aux opérations de dimensionnement automatique et de redimensionnement manuel. Pour en savoir plus sur cette configuration, consultez Réduction de capacité des clusters.

Les paramètres suivants déterminent le comportement de dimensionnement automatique pour chaque règle d'une stratégie.

> **Note**
>
> Les paramètres répertoriés ici sont basés sur ceux AWS Management Console d'Amazon EMR. Lorsque vous utilisez l'API AWS CLI ou Amazon EMR, des options de configuration avancées supplémentaires sont disponibles. Pour plus d'informations sur les options avancées, consultez SimpleScalingPolicyConfigurationle manuel Amazon EMR API Reference.

- Nombre minimal et nombre maximal d'instances. La contrainte 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 mise à l'échelle (monter en puissance) De même, la contrainte Nombre minimal d'instances spécifie le nombre minimal d'instances Amazon EC2 et s'applique à toutes les règles de mise à l'échelle (mise à l'échelle horizontale).
- 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 CloudWatch métrique, qui est surveillée pour détecter une situation d'alarme.
- Un opérateur de comparaison, qui est utilisé pour comparer la CloudWatch métrique à la valeur du seuil et déterminer une condition de déclenchement.
- Période d'évaluation, par tranches de cinq minutes, pendant laquelle la CloudWatch métrique doit être dans une condition de déclenchement avant que l'activité de dimensionnement ne soit déclenchée.
- 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. Lorsqu'un groupe d'instances a terminé une activité de dimensionnement et atteint son état après le dimensionnement, la période de recharge permet aux CloudWatch métriques susceptibles de déclencher les activités de dimensionnement ultérieures de se stabiliser. Pour de plus amples informations, consultez Temps de stabilisation de l'autoscaling dans le Guide de l'utilisateur pour Amazon EC2 Auto Scaling.



Considérations et restrictions

- Les CloudWatch métriques Amazon sont essentielles au bon fonctionnement du dimensionnement automatique d'Amazon EMR. Nous vous recommandons de suivre de près les CloudWatch métriques Amazon pour vous assurer que les données ne sont pas manquantes. Pour plus d'informations sur la façon dont vous pouvez configurer les CloudWatch alarmes Amazon afin de détecter les métriques manquantes, consultez Utilisation des CloudWatch alarmes Amazon.
- La surutilisation des volumes EBS peut entraîner des problèmes de mise à l'échelle gérée. Nous vous recommandons de surveiller de près l'utilisation du volume EBS pour vous assurer que le volume EBS est inférieur à 90 % d'utilisation. Consultez Stockage d'instance pour plus d'informations sur la spécification de volumes EBS supplémentaires.

- Le dimensionnement automatique avec une politique personnalisée dans les versions 5.18 à 5.28 d'Amazon EMR peut rencontrer un échec de dimensionnement dû à l'absence intermittente de données dans les métriques Amazon. CloudWatch Pour une mise à l'échelle automatique améliorée, nous vous recommandons d'utiliser les versions les plus récentes d'Amazon EMR. Si vous avez besoin d'utiliser une version d'Amazon EMR comprise entre 5.18 et 5.28, vous pouvez contacter l'AWS Assistance pour obtenir un correctif.

Utilisation du AWS Management Console pour configurer le dimensionnement automatique

Lorsque vous créez un cluster, vous configurez une politique de mise à l'échelle pour les groupes d'instances à l'aide des options de configuration avancée du cluster. 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.

> **Note**
>
> La nouvelle console Amazon EMR (https://console.aws.amazon.com/emr) utilise la mise à l'échelle gérée au lieu de l'autoscaling. Pour utiliser l'autoscaling, assurez-vous d'être connecté à l'ancienne console à l'adresse https://console.aws.amazon.com/elasticmapreduce.

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Si vous créez un cluster, dans la console Amazon EMR, sélectionnez Créer un cluster, sélectionnez Accéder aux options avancées, choisissez des options pour Étape 1 : Logiciel et étapes, puis accédez 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).
3. Dans l'option de mise à l'échelle et mise en service du cluster, sélectionnez Activer la mise à l'échelle 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 d'autoscaling s'ouvre.

4. 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.
5. 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.
6. Choisissez les paramètres de règle comme indiqué précédemment dans cette rubrique. Pour obtenir une description des CloudWatch métriques disponibles pour Amazon EMR, consultez les métriques et dimensions Amazon EMR dans le guide de l'utilisateur Amazon. CloudWatch

Utilisation du AWS CLI pour configurer le dimensionnement automatique

Vous pouvez utiliser AWS CLI des commandes pour Amazon EMR afin de configurer le dimensionnement automatique lorsque vous créez un cluster et 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 doté d'une politique de mise à l'échelle automatique, vous devez utiliser la commande `--auto-scaling-role` *`MyAutoScalingRole`* pour spécifier le rôle IAM pour la 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 politique de dimensionnement automatique, consultez le manuel PutAutoScalingPolicyAmazon EMR API Reference.

Création d'un cluster avec une politique de mise à l'échelle 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 équivalente à la politique de scale-out par défaut qui apparaît lorsque vous créez une politique de dimensionnement automatique avec le for AWS Management Console 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
scale-out,Description=Replicates the default scale-out rule in the
 console.,Action={SimpleScalingPolicyConfiguration={AdjustmentType=CHANGE_IN_CAPACITY,ScalingAd
ElasticMapReduce,Period=300,Statistic=AVERAGE,Threshold=15,Unit=PERCENT,Dimensions=[{Key=JobFlo
```

La commande suivante illustre comment utiliser la ligne de commande pour fournir la définition de la politique de mise à l'échelle 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 politique de mise à l'échelle automatique à un cluster

Vous pouvez spécifier une configuration de la politique de mise à l'échelle à l'aide de l'option `--instance-groups` avec la commande `add-instance-groups` de la même manière 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 politique de mise à l'échelle automatique à un groupe d'instances existant ou modification d'une politique 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 mise à l'échelle 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": {
                  "MaxCapacity": 10,
                  "MinCapacity": 2
          },
          "Rules": [{
                  "Action": {
```

```
                        "SimpleScalingPolicyConfiguration": {
                                "AdjustmentType": "CHANGE_IN_CAPACITY",
                                "CoolDown": 300,
                                "ScalingAdjustment": 1
                        }
                },
                "Description": "Replicates the default scale-out rule in the console
for YARN memory",
                "Name": "Default-scale-out",
                "Trigger": {
                        "CloudWatchAlarmDefinition": {
                                "ComparisonOperator": "LESS_THAN",
                                "Dimensions": [{
                                        "Key": "JobFlowID",
                                        "Value": "${emr.clusterID}"
                                }],
                                "EvaluationPeriods": 1,
                                "MetricName": "YARNMemoryAvailablePercentage",
                                "Namespace": "AWS/ElasticMapReduce",
                                "Period": 300,
                                "Statistic": "AVERAGE",
                                "Threshold": 15,
                                "Unit": "PERCENT"
                        }
                }
        }]
 }
```

Suppression d'une politique de mise à l'échelle automatique d'un groupe d'instances

```
aws emr remove-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07
```

Récupération de la configuration d'une politique de mise à l'échelle automatique

La `describe-cluster` commande récupère la configuration de la politique dans le InstanceGroup bloc. 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 de groupes d'instances principaux et de tâches et de flottes d'instances dans un cluster en cours d'exécution avec l' AWS CLI API AWS Management Console, ou 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 Options de parc d'instances. 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 met fin aux tâches sans interrompre les autres, et prévient la perte de données. Pour plus d'informations, consultez Mise hors service lors de l'achèvement de la tâche.

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.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Modifier le nombre d'instances d'un cluster existant à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2, dans le volet de navigation de gauche, choisissez Clusters, puis sélectionnez le cluster que vous souhaitez mettre à jour. Le cluster doit être en cours d'exécution ; vous ne pouvez pas redimensionner un cluster de en cours de mise en service ou un cluster arrêté.
3. Dans l'onglet Instances de la page de détails du cluster, consultez le panneau Groupes d'instances.
4. Pour redimensionner un groupe d'instances existant, sélectionnez le bouton radio à côté du groupe d'instances principal ou de tâches que vous souhaitez redimensionner, puis choisissez Redimensionner le groupe d'instances. Spécifiez le nouveau nombre d'instances pour le groupe d'instances, puis sélectionnez Redimensionner.

   > **Note**
   >
   > Si vous choisissez de réduire la taille d'un groupe d'instances en cours d'exécution, Amazon EMR sélectionnera intelligemment les instances à supprimer du groupe afin de minimiser les pertes de données. Pour un contrôle plus précis de votre action de

redimensionnement, vous pouvez sélectionner l'ID du groupe d'instances, choisir les instances que vous souhaitez supprimer, puis utiliser l'option Terminer. Pour plus d'informations sur le comportement intelligent de réduction, consultez Réduction de capacité des clusters.

5. Si vous souhaitez annuler l'action de redimensionnement, vous pouvez sélectionner le bouton radio correspondant à un groupe d'instances ayant le statut Redimensionner, puis choisir Arrêter le redimensionnement dans la liste des actions.
6. Pour ajouter un ou plusieurs groupes d'instances de tâches à votre cluster en réponse à l'augmentation de la charge de travail, choisissez Ajouter un groupe d'instances de tâches dans la liste des actions. Choisissez le type d'instance Amazon EC2, entrez le nombre d'instances pour le groupe de tâches, puis sélectionnez Ajouter un groupe d'instances de tâches pour revenir au panneau Groupes d'instances de votre cluster.

Old console

Modifier le nombre d'instances d'un cluster existant à l'aide de l'ancienne 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 flottes d'instances, choisissez Redimensionner dans la colonne Capacité allouée, saisissez de nouvelles valeurs pour les unités à la demande et les 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.

Redimensionnez un cluster à l'aide du AWS CLI

Vous pouvez utiliser le 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 d'arrêter une instance du groupe d'instances principal à l'aide de l'API AWS CLI ou de l'API. Cela doit être effectué avec précautions. L'arrêt d'une instance dans le groupe d'instances principal entraîne un risque de perte de 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'interface AWS CLI.

Pour redimensionner un cluster en modifiant le nombre d'instances à l'aide du 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 AWS CLI `modify-instance-groups` sous-commande associée au `InstanceCount` paramètre. Pour ajouter des instances aux groupes principaux ou de tâches, augmentez la valeur `InstanceCount`. Pour réduire le nombre d'instances dans le groupe de tâches, diminuez la valeur `InstanceCount`. La mise à zéro du nombre d'instances du groupe de tâches supprime toutes les instances mais pas le groupe d'instances.

- Pour augmenter le nombre d'instances du groupe d'instances de tâches de 3 à 4, tapez la commande suivante et remplacez *`ig-31JXXXXXXBT0`* par l'ID du groupe d'instances.

  ```
  aws emr modify-instance-groups --instance-groups
   InstanceGroupId=ig-31JXXXXXXBT0,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 de groupe d'instances, tapez la commande suivante et remplacez *`j-2AXXXXXXGAPLF`* par l'ID du cluster.

  ```
  aws emr describe-cluster --cluster-id j-2AXXXXXXGAPLF
  ```

  Avec le AWS CLI, vous pouvez également mettre fin à une instance du groupe d'instances principal à l'aide de la `--modify-instance-groups` sous-commande.

  > **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 arrêter une instance spécifique, vous avez besoin de l'ID du groupe d'instances (retourné par la sous-commande `aws emr describe-cluster --cluster-id`) et de l'ID d'instance (renvoyé par la sous-commande `aws emr list-instances --cluster-id`). Tapez la commande suivante, remplacez *ig-6RXXXXXX07SA* par l'ID du groupe d'instances et remplacez *i-f9XXXXf2* par 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Pour redimensionner un cluster en ajoutant des groupes d'instances de tâches avec AWS CLI

Avec le AWS CLI, vous pouvez ajouter de 1 à 48 groupes d'instances de tâches à un cluster à l'aide de la `--add-instance-groups` sous-commande. Les groupes d'instances de tâches peuvent uniquement être ajoutés à un cluster contenant un groupe d'instances primaire et un groupe d'instances principal. Lorsque vous utilisez le AWS CLI, vous pouvez ajouter jusqu'à cinq groupes d'instances de tâches à chaque fois que vous utilisez la `--add-instance-groups` sous-commande.

1. Pour ajouter un groupe d'instances de tâches unique à un cluster, tapez la commande suivante et remplacez *j-JXBXXXXXX37R* par l'ID du 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* par l'ID du 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 le 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 du AWS CLI

Vous pouvez utiliser le AWS CLI pour arrêter un redimensionnement à l'aide de la `modify-instance-groups` sous-commande. 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 Interrompu

Un groupe d'instances passe à l'état interrompu s'il rencontre trop d'erreurs en essayant de démarrer les nouveaux nœuds du cluster. Par exemple, si les nouveaux nœuds échouent lors de la réalisation d'actions d'amorçage, le groupe d'instances passe à l'état INTERROMPU au lieu de mettre en service en permanence 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 le AWS CLI, la `list-instances` sous-commande renvoie toutes les instances et leurs états, tout comme la `describe-cluster` sous-commande. Si Amazon EMR détecte un défaut avec un groupe d'instances, il change l'état du groupe en spécifiant SUSPENDED.

Pour remettre un cluster dans un état SUSPENDU à l'aide du 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 les groupes d'instances d'un cluster, tapez la commande suivante et remplacez *`j-3KVXXXXXXY7UG`* par l'ID du 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-3SUXXXXXQ9ZM",
                "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 primaires, 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-3SUXXXXXQ9ZM,InstanceCount=3
```

Considérations relatives à la réduction de la taille du cluster

Si vous choisissez de réduire la taille d'un cluster en cours d'exécution, tenez compte du comportement et des meilleures pratiques d'Amazon EMR suivantes :

- Pour réduire l'impact sur les tâches en cours, Amazon EMR sélectionne intelligemment les instances à supprimer. Pour plus d'informations sur le comportement de réduction des clusters, consultez Mise hors service lors de l'achèvement de la tâche dans le Guide de gestion Amazon EMR.

- Lorsque vous réduisez la taille d'un cluster, Amazon EMR copie les données des instances qu'il supprime vers les instances restantes. Assurez-vous que la capacité de stockage de ces données est suffisante dans les instances qui restent dans le groupe.
- Amazon EMR tente de mettre hors service HDFS sur les instances du groupe. Avant de réduire la taille d'un cluster, nous vous recommandons de minimiser les E/S d'écriture HDFS.
- Pour un contrôle plus précis lorsque vous réduisez la taille d'un cluster, vous pouvez afficher le cluster dans la console et accéder à l'onglet Instances. Sélectionnez l'ID du groupe d'instances que vous souhaitez redimensionner. Utilisez ensuite l'option Terminer pour les instances spécifiques que vous souhaitez supprimer.

Configurer les délais d'expiration pour la mise en service de la capacité

Lorsque vous utilisez des flottes d'instances, vous pouvez configurer les délais d'expiration de la mise en service. Un délai de mise en service indique à Amazon EMR d'arrêter la mise en service de la capacité de l'instance si le cluster dépasse un seuil de temps spécifié lors du lancement du cluster ou des opérations de mise à l'échelle du cluster. Les rubriques suivantes expliquent comment configurer un délai de mise en service pour le lancement du cluster et pour les opérations d'augmentation du cluster.

Rubriques

- Configurer les délais de mise en service pour le lancement du cluster dans Amazon EMR
- Personnaliser le délai d'expiration de la mise en service pour le redimensionnement du cluster dans Amazon EMR

Configurer les délais de mise en service pour le lancement du cluster dans Amazon EMR

Vous pouvez définir un délai d'expiration pour fournir des instances Spot pour chaque flotte de votre cluster. Si Amazon EMR ne peut pas mettre en service la capacité Spot, vous pouvez choisir de mettre fin au cluster ou de mettre en service la capacité à la demande à la place. Si le délai expire pendant le processus de redimensionnement du cluster, Amazon EMR annule les demandes Spot qui n'ont pas été mises en service. Les instances Spot qui n'ont pas été mises en service ne sont pas transférées vers des capacités à la demande.

> **Note**
>
> Vous ne pouvez pas personnaliser le délai d'expiration de la mise en service dans l'ancienne console. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

Pour personnaliser le délai de mise en service pour le lancement du cluster avec la console Amazon EMR, procédez comme suit.

New console

Configurer le délai d'expiration de la mise en service lorsque vous créez un cluster avec la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.
3. Sur la page Créer un cluster, accédez à Configuration du cluster et sélectionnez Flottes d'instances.
4. Sous l'option de mise à l'échelle et de mise en service du cluster, spécifiez la taille Spot pour votre nœud principal et vos flottes de tâches.
5. Sous Configuration du délai d'expiration Spot, sélectionnez Terminer le cluster après un délai d'expiration Spot ou Passer sur À la demande après un délai d'expiration Spot. Spécifiez ensuite le délai d'expiration pour la mise en service des instances Spot. La valeur par défaut est 1 heure.
6. Choisissez toutes les autres options qui s'appliquent à votre cluster.
7. Pour lancer votre cluster avec le délai d'expiration configuré, choisissez Créer un cluster.

AWS CLI

Spécifier un délai de mise en service à l'aide de la commande **`create-cluster`**

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
```

```
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSp
{"OnDemandSpecification":{"AllocationStrategy":"lowest-
price"}},"InstanceTypeConfigs":[{"WeightedCapacity":1,"EbsConfiguration":
{"EbsBlockDeviceConfigs":[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecifi
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification"
{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 2"}]'
```

Personnaliser le délai d'expiration de la mise en service pour le redimensionnement du cluster dans Amazon EMR

Vous pouvez définir un délai d'expiration pour mettre en service des instances Spot pour chaque flotte de votre cluster. Si Amazon EMR ne parvient pas à fournir la capacité Spot, il annule la demande de redimensionnement et arrête d'essayer de fournir de la capacité Spot supplémentaire. Lorsque vous créez un cluster, vous pouvez créer un délai d'expiration. Pour un cluster en cours d'exécution, vous pouvez ajouter ou mettre à jour un délai d'expiration.

Lorsque le délai d'expiration expire, Amazon EMR envoie automatiquement les événements vers un flux CloudWatch Amazon Events. Avec CloudWatch, vous pouvez créer des règles qui correspondent aux événements selon un modèle spécifié, puis acheminer les événements vers des cibles pour qu'ils prennent des mesures. Par exemple, vous pouvez créer une règle pour envoyer une notification par e-mail. Pour plus d'informations sur la création de règles, consultez Création de règles pour les événements Amazon EMR avec CloudWatch. Pour plus d'informations sur les différents détails de l'événement, consultez Événements de modification de l'état de la flotte d'instances.

Exemples de délais de mise en service pour le redimensionnement du cluster

Spécifiez un délai de mise en service pour le redimensionnement à l'aide de l'interface AWS CLI

L'exemple suivant utilise la commande `create-cluster` pour ajouter un délai de mise en service pour le redimensionnement.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTyp
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

L'exemple suivant utilise la commande `modify-instance-fleet` pour ajouter un délai de mise en service pour le redimensionnement.

```
aws emr modify-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet '{"InstanceFleetId":"if-XXXXXXXXXXX","ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":60}}}' \
--region us-east-1
```

L'exemple suivant utilise le `add-instance-fleet-`command pour ajouter un délai de mise en service pour le redimensionnement.

```
aws emr add-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet
 '{"InstanceFleetType":"TASK","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTypeCo
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
```

```
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":35}}}' \
--region us-east-1
```

Spécifiez un délai d'approvisionnement pour le redimensionnement et le lancement à l'aide du AWS CLI

L'exemple suivant utilise la commande `create-cluster` pour ajouter un délai de mise en service pour le redimensionnement et le lancement.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSpeci
{"OnDemandSpecification":{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

Considérations relatives au redimensionnement des délais de mise en service

Lorsque vous configurez les délais d'expiration de la mise en service des clusters pour vos flottes d'instances, tenez compte des comportements suivants.

- Vous pouvez créer des délais de mise en service pour des instances Spot et à la demande. Le délai de mise en service minimal est de 5 minutes. Le délai de mise en service maximal est de 7 jours.

- Vous ne pouvez configurer les délais de mise en service que pour un cluster EMR qui utilise des flottes d'instances. Vous devez configurer chaque flotte de tâches et nœud principal séparément.
- Lorsque vous créez un cluster, vous pouvez créer des délais de mise en service. Vous pouvez ajouter un délai d'expiration ou mettre à jour un délai d'expiration existant pour un cluster en cours d'exécution.
- Si vous soumettez plusieurs opérations de redimensionnement, Amazon EMR suit les délais de mise en service pour chaque opération de redimensionnement. Par exemple, définissez le délai de mise en service d'un cluster sur *60* minutes. *Soumettez ensuite une opération de redimensionnement R1 à l'instant T1.* Soumettez une deuxième opération de redimensionnement *R2* à l'instant *T2*. Le délai de mise en service pour R1 expirera à *T1 +60 minutes*. Le délai de mise en service pour R2 expirera à *T2 + 60 minutes*.
- Si vous soumettez une nouvelle opération d'augmentation avant l'expiration du délai imparti, Amazon EMR continue d'essayer de fournir de la capacité pour votre cluster EMR.

Réduction de capacité des clusters

> **Note**
>
> Les options de comportement de réduction ne sont plus prises en charge depuis la version 5.10.0 d'Amazon EMR. En raison de l'introduction de la facturation à la seconde dans Amazon EC2, le comportement de réduction de capacité par défaut pour les clusters Amazon EMR se termine désormais à la fin de la tâche.

Avec les versions 5.1.0 à 5.9.1 d'Amazon EMR, il existe deux options pour le comportement de réduction 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 le AWS CLI pour émettre un `modify-instance-groups` avec`EC2InstanceIdsToTerminate`, ces instances sont immédiatement résiliées, sans tenir compte de ces paramètres, et quel que soit le statut des applications qui y sont

exécutées. Une telle résiliation d'une instance peut entraîner une perte des données et un comportement du cluster imprévisible.

Lorsque l'option Terminer à l'achèvement de la tâche est spécifiée, Amazon EMR place les tâches sur liste noire et les vide des nœuds avant de mettre fin aux 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 réduire votre cluster sans attribuer votre charge de travail. Lors d'une opération de redimensionnement, Amazon EMR met gracieusement hors service les démons YARN, HDFS et autres sur les nœuds principaux et les nœuds de tâches, le tout sans perdre de données ni interrompre les tâches. Amazon EMR ne réduit la taille des groupes d'instances que si le travail assigné aux groupes est terminé et qu'ils sont inactifs. Pour YARN NodeManager Graceful Decommission, vous pouvez ajuster manuellement le temps d'attente d'un nœud avant la mise hors service.

Cette durée est définie à l'aide d'une propriété dans la classification de configuration `YARN-site`. Avec Amazon EMR versions 5.12.0 et ultérieures, 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 attribués sur d'autres nœuds. La valeur par défaut est de 3 600 secondes (une heure). Vous pouvez attribuer 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'ont pas de tâches en cours d'exécution pour une étape ou une application, et supprime d'abord ces instances 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 de 1 heure. Cette valeur peut être modifiée à l'aide du paramètre `YARN.resourcemanager.decommissioning.timeout`. Amazon EMR utilise dynamiquement le nouveau paramètre. Vous pouvez définir ce paramètre sur un nombre arbitrairement élevé pour garantir qu'Amazon EMR ne met fin à aucune tâche tout en réduisant la taille du cluster.

Groupes de nœuds principaux

Sur les nœuds principaux, les DataNode démons YARN NodeManager et HDFS 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 devez également essayer de minimiser les E/S d'écriture avant de tenter de réduire les groupes d'instances. Des E/S d'écriture excessives peuvent retarder la fin de l'opération de redimensionnement.

Une autre limite est le facteur de réplication par défaut, `dfs.replication` dans `/etc/hadoop/conf/hdfs-site`. Lorsqu'il crée un cluster, 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.

⚠ Warning

1. Paramétrer `dfs.replication` sur la valeur 1 avec les clusters de moins de quatre nœuds peut entraîner une perte de données HDFS en cas de panne d'un seul nœud. Nous vous recommandons d'utiliser un cluster comportant au moins quatre nœuds principaux pour les charges de travail de production.
2. Amazon EMR n'autorisera pas les clusters à mettre à l'échelle les nœuds principaux situés en dessous de `dfs.replication`. Par exemple, si `dfs.replication = 2`, le nombre minimum de nœuds principaux est 2.

3. Lorsque vous utilisez la mise à l'échelle gérée, autoscaling, ou que vous choisissez de redimensionner manuellement votre cluster, nous vous recommandons de définir `dfs.replication` sur une valeur supérieure ou égale à 2.

La réduction progressive ne vous permet pas de réduire les nœuds principaux en dessous du facteur de réplication HDFS. Cela permet à HDFS de fermer des fichiers en raison d'un nombre insuffisant de répliques. Pour contourner cette limite, réduisez le facteur de réplication et redémarrez le NameNode daemon.

Configurer le comportement de réduction d'Amazon EMR

> **Note**
>
> L'option de réduction du comportement de résiliation à l'heure de l'instance n'est plus prise en charge pour les versions 5.10.0 et ultérieures d'Amazon EMR. Les options de comportement de réduction suivantes apparaissent uniquement dans la console Amazon EMR pour les versions 5.1.0 à 5.9.1.

Vous pouvez utiliser l'API AWS Management Console AWS CLI, la ou l'API Amazon EMR pour configurer le comportement de réduction lorsque vous créez un cluster.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Consultez Console Amazon EMR pour en savoir plus sur les différences entre l'ancienne et la nouvelle expérience console.

New console

Configurer le comportement de réduction à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters, puis Créer un cluster.

3. Dans la section Options de mise à l'échelle et de mise en service du cluster, recherchez Terminaison du cluster et choisissez de terminer manuellement votre cluster ou de demander à Amazon EMR de le terminer après une durée d'inactivité spécifiée. Activez éventuellement la protection de terminaison contre les bogues ou les erreurs.
4. Choisissez toutes les autres options qui s'appliquent à votre cluster.
5. Pour lancer cluster, choisissez Créer un cluster.

Old console

Configurer le comportement de réduction à l'aide de l'ancienne console

1. Ouvrez la console Amazon EMR à l'adresse https://console.aws.amazon.com/elasticmapreduce.
2. Choisissez Créer un cluster. Accédez aux options avancées et choisissez vos paramètres de configuration à l'Étape 1 : Logiciel et étapes et à l'Étape 2 : Matériel.
3. À l'Étape 3 : Paramètres généraux du cluster, sélectionnez le comportement de réduction souhaité. Complétez les configurations restantes et créez votre cluster.

AWS CLI

Pour configurer le comportement de réduction à l'aide du AWS CLI

- Utilisez l'option `--scale-down-behavior` pour spécifier `TERMINATE_AT_INSTANCE_HOUR` ou `TERMINATE_AT_TASK_COMPLETION`.

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 Contrôle de la mise hors service d'un cluster. 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 souhaitez résilier 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 interrompus à l'aide de la console AWS CLI, de ou par programmation à l'aide de l'`TerminateJobFlows`API.

En fonction de la configuration du cluster, il peut falloir de 5 à 20 minutes au cluster pour se résilier totalement et libérer les ressources allouées, telles que des instances EC2.

> **Note**
>
> Vous ne pouvez pas redémarrer un cluster arrêté, mais vous pouvez le cloner pour réutiliser sa configuration pour un nouveau cluster. Pour plus d'informations, consultez Clonage d'un cluster à l'aide de la console.

> **Important**
>
> Amazon EMR utilise la fonction du service Amazon EMR et le `AWSServiceRoleForEMRCleanup` rôle pour nettoyer les ressources de cluster de votre compte que vous n'utilisez plus, telles que les instances Amazon EC2. Vous devez inclure des actions pour les politiques de rôle afin de supprimer ou de résilier les ressources. Dans le cas contraire, Amazon EMR ne pourra pas effectuer ces actions de nettoyage, et les ressources non utilisées qui restent sur le cluster risquent de générer des coûts.

Résilier 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.

New console

Arrêt d'un cluster à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Choisissez Clusters, puis sélectionnez le cluster que vous voulez arrêter.
3. Dans le menu déroulant Actions, choisissez Arrêter un cluster pour ouvrir l'invite Arrêter le cluster.

4. À l'invite, choisissez Arrêter. Selon la configuration du cluster, l'arrêt peut prendre entre 5 à 10 minutes. Pour plus d'informations sur la création de clusters Amazon EMR, consultez Arrêter un cluster.

Old console

Mettre fin à un cluster dont la protection contre la terminaison est désactivée à l'aide de l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Sélectionnez le cluster à arrêter. Vous pouvez sélectionner plusieurs clusters et les suspendre simultanément.
3. Sélectionnez Résilier.
4. A l'invite, choisissez Résilier.

   Amazon EMR arrête les instances dans le cluster et arrête d'enregistrer des données de journal.

Mettre fin à un cluster dont la protection contre la terminaison est activée à l'aide de l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Sur la page Liste de clusters, sélectionnez le cluster à arrêter. Vous pouvez sélectionner plusieurs clusters et les suspendre simultanément.
3. Sélectionnez Résilier.
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 arrête les instances dans le cluster et arrête d'enregistrer des données de journal.

Résilier un cluster à l'aide de l' AWS CLI

Pour mettre fin à un cluster non protégé à l'aide du AWS CLI

Pour arrêter un cluster non protégé à l'aide de AWS CLI, utilisez la `terminate-clusters` sous-commande 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Pour mettre fin à un cluster protégé à l'aide du AWS CLI

Pour arrêter un cluster protégé à l'aide de la AWS CLI, désactivez d'abord la protection de terminaison à l'aide de la `modify-cluster-attributes` sous-commande avec le `--no-termination-protected` paramètre. 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 le AWS CLI, consultez. https://docs.aws.amazon.com/cli/latest/reference/emr

Résilier un cluster à l'aide de l'API

L'opération `TerminateJobFlows` termine le traitement de l'étape, charge toutes données de journal d'Amazon EC2 vers Amazon S3 (si configuré) et arrête 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 plus d'informations sur les paramètres d'entrée uniques à`TerminateJobFlows`, consultez TerminateJobFlows. Pour de plus amples informations sur les paramètres génériques dans la demande, consultez Paramètres de demande communs.

Clonage d'un cluster à l'aide de la 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.

> **Note**
>
> Nous avons repensé la console Amazon EMR pour en faciliter l'utilisation. Dans la nouvelle console, vous pouvez cloner des clusters qui utilisent le dimensionnement automatique.

Cependant, vous ne pouvez créer de nouveaux clusters que si vous souhaitez les mettre à l'échelle manuellement ou utiliser le dimensionnement géré. Consultez Console Amazon EMR pour en savoir plus sur les différences entre les anciennes et les nouvelles expériences de console.

New console

Cloner un cluster à l'aide de la nouvelle console

1. Connectez-vous à la AWS Management Console console Amazon EMR et ouvrez-la à l'adresse https://console.aws.amazon.com/emr.
2. Sous EMR sur EC2 dans le volet de navigation de gauche, choisissez Clusters.
3. Cloner un cluster à partir de la liste des clusters
   a. Utilisez les options de recherche et de filtrage pour trouver le cluster que vous souhaitez cloner dans la vue de liste.
   b. Cochez la case à gauche de la ligne pour le cluster que vous souhaitez cloner.
   c. L'option Cloner sera désormais disponible en haut de la liste. Sélectionnez Cloner pour lancer le processus de clonage. Si le cluster possède des étapes configurées, choisissez Inclure les étapes et Continuer si vous souhaitez cloner les étapes avec les autres configurations de cluster.
   d. Vérifiez les paramètres du nouveau cluster qui ont été copiés depuis le cluster cloné. Ajustez les paramètres si nécessaire. Lorsque vous êtes satisfait de la configuration du nouveau cluster, sélectionnez Créer un cluster pour lancer le nouveau cluster.
4. Cloner un cluster à partir de la page de détails du cluster
   a. Pour accéder à la page de détails du cluster que vous souhaitez cloner, sélectionnez son ID de cluster dans la vue de liste des clusters.
   b. En haut de la page de détails du cluster, sélectionnez Cloner le cluster dans le menu Actions pour lancer le processus de clonage. Si le cluster possède des étapes configurées, choisissez Inclure les étapes et Continuer si vous souhaitez cloner les étapes avec les autres configurations de cluster.
   c. Vérifiez les paramètres du nouveau cluster qui ont été copiés depuis le cluster cloné. Ajustez les paramètres si nécessaire. Lorsque vous êtes satisfait de la configuration du nouveau cluster, sélectionnez Créer un cluster pour lancer le nouveau cluster.

Old console

Cloner un cluster à l'aide de l'ancienne console

1. Accédez à la nouvelle console Amazon EMR et sélectionnez Changer pour l'ancienne console depuis le menu latéral. Pour plus d'informations sur ce qu'implique le passage à l'ancienne console, consultez la rubrique Utilisation de l'ancienne console.
2. Choisissez Créer un cluster.
3. Sur la page Liste de clusters, cliquez sur un cluster à cloner.
4. 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 Soumission de travail à un cluster.
5. 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).

Automatisation de clusters récurrents avec AWS Data Pipeline

AWS Data Pipeline est un service qui automatise le mouvement et la transformation des 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 l'utiliser AWS Data Pipeline pour planifier ces clusters. AWS Data Pipeline est un flux de travail piloté par les données, de sorte qu'une tâche (lancement du cluster) peut dépendre d'une autre tâche (déplacement des données d'entrée vers Amazon S3). Il possède également une puissante fonctionnalité pour les nouvelles tentatives.

Pour plus d'informations AWS Data Pipeline, consultez le guide du AWS Data Pipeline développeur, en particulier les didacticiels concernant Amazon EMR :

- Didacticiel : Lancement d'un flux de travail Amazon EMR
- Mise en route : traitez les journaux Web avec AWS Data Pipeline Amazon EMR et Hive
- Tutoriel : importation et exportation d'Amazon DynamoDB à l'aide de AWS Data Pipeline

Résolution des problèmes liés à un cluster

Un cluster EMR s'exécute dans un écosystème complexe comprenant des logiciels open source, du code d'application personnalisé et. Services AWS Lorsqu'un problème survient avec l'une de ces parties, le cluster peut échouer ou prendre plus de temps que prévu à s'arrêter. Les rubriques suivantes peuvent vous aider à identifier les problèmes de cluster et à les résoudre.

Rubriques

- Quels sont les outils disponibles pour résoudre les problèmes ?
- Afficher et redémarrer Amazon EMR et les processus d'application (démon)
- Erreurs courantes dans Amazon EMR
- Dépannage d'un cluster ayant échoué
- Résolution des problèmes de rapidité d'un cluster
- Résoudre les problèmes liés à un cluster de Lake Formation

Lorsque vous développez une nouvelle application Hadoop, nous vous recommandons d'activer le débogage et de traiter un sous-ensemble restreint, mais représentatif de vos données pour tester l'application. Vous pouvez également exécuter l'application step-by-step pour tester chaque étape séparément. Pour plus d’informations, consultez Configuration de la journalisation et du débogage du cluster et Étape 5 : Test du cluster étape par étape.

Quels sont les outils disponibles pour résoudre les problèmes ?

Pour identifier et corriger les erreurs de cluster, vous pouvez utiliser les outils décrits sur cette page. Lorsque vous lancez le cluster, il se peut que vous deviez initialiser certains outils. D'autres outils sont disponibles par défaut pour chaque cluster.

Rubriques

- Consulter les détails du cluster EMR
- Afficher les détails des erreurs du cluster EMR
- Exécuter des scripts et configurer les processus Amazon EMR
- Afficher les fichiers journaux
- Surveillez les performances du cluster EMR

Consulter les détails du cluster EMR

Vous pouvez utiliser l'API AWS Management Console AWS CLI, ou EMR pour récupérer des informations détaillées sur un cluster EMR et l'exécution des tâches. Pour plus d'informations sur l'utilisation du AWS Management Console et AWS CLI, consultezAfficher l'état et les détails d'un cluster.

Volet de détails de la console Amazon EMR

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 AWS. La liste affiche tous les clusters actifs et terminés que vous avez lancés au cours des deux derniers mois. 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 disponibles dans la page de détails du cluster peuvent être utiles pour dépanner les clusters. 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. Cette fonctionnalité n'est disponible que pour les versions 5.8.0 et supérieures d'Amazon EMR.

Interface de ligne de commande Amazon EMR

Vous pouvez trouver des informations sur un cluster à l' AWS CLI aide de l'`--describe`argument.

API Amazon EMR

Vous pouvez rechercher les détails relatifs à un cluster à partir de l'API à l'aide de l'action `DescribeJobFlows`.

Afficher les détails des erreurs du cluster EMR

Lorsqu'un cluster EMR s'arrête avec une erreur, les API `DescribeCluster` et `ListClusters` renvoient un code d'erreur et un message d'erreur. Pour certaines erreurs de cluster, le tableau de données `ErrorDetail` peut vous aider à résoudre le problème.

Pour obtenir la liste des codes d'erreur incluant des données `ErrorDetail`, consultez Codes d'erreur avec ErrorDetail informations.

> **Note**
>
> Nous affinons continuellement nos messages d'erreur afin que vous receviez les informations les plus récentes et les plus pertinentes. Nous vous déconseillons d'analyser le texte à partir de `ErrorMessage`, car celui-ci est sujet à modification.

Exécuter des scripts et configurer les processus Amazon EMR

Dans le cadre de votre processus de résolution des problèmes, il peut être utile d'exécuter des scripts personnalisés sur votre cluster ou d'afficher et de configurer les processus du cluster.

Afficher et redémarrer les processus d'application

Il peut être utile de visualiser les processus en cours sur votre cluster afin de diagnostiquer les problèmes potentiels. Vous pouvez arrêter et redémarrer les processus du cluster en vous connectant au nœud principal de votre cluster. Pour plus d'informations, consultez Afficher et redémarrer Amazon EMR et les processus d'application (démon).

Exécuter des commandes et des scripts sans connexion SSH

Pour exécuter une commande ou un script sur votre cluster en tant qu'étape, vous pouvez utiliser les outils `command-runner.jar` ou `script-runner.jar` sans établir de connexion SSH avec le nœud principal. Pour plus d'informations, consultez Exécuter des commandes et des scripts sur un cluster Amazon EMR.

Afficher les fichiers journaux

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 Configuration de la journalisation et du débogage du cluster.

Fichiers journaux sur le nœud principal

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.

Fichiers journaux archivés sur 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.

Surveillez les performances du cluster EMR

Amazon EMR propose plusieurs outils pour surveiller les performances de votre cluster.

Interfaces Web Hadoop

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.

CloudWatch métriques

Chaque cluster communique des métriques à CloudWatch. CloudWatch est un service Web qui suit les métriques et que vous pouvez utiliser pour définir des alarmes sur ces métriques. Pour plus d’informations, consultez Surveillance des métriques Amazon EMR avec CloudWatch.

Afficher et redémarrer Amazon EMR et les processus d'application (démon)

Lorsque vous dépannez un cluster, vous pouvez souhaiter afficher les processus en cours d'exécution. Il se peut que vous ayez besoin de redémarrer ou arrêter des processus. Par exemple, vous pouvez redémarrer un processus après avoir modifié une configuration ou remarquer un problème avec un processus particulier, une fois que vous avez analysé les fichiers journaux et les messages d'erreur.

Deux types de processus s'exécutent sur un cluster : les processus Amazon EMR (par exemple, instance-controller et Log Pusher) et les processus associés aux applications installées sur le cluster (par exemple, et). hadoop-hdfs-namenode hadoop-yarn-resourcemanager

Pour utiliser directement les processus sur un cluster, vous devez d'abord vous connectez au nœud principal. Pour plus d'informations, consultez Connexion à un cluster.

Affichage des processus en cours d'exécution

La méthode pour visualiser les processus en cours sur un cluster varie en fonction de la version d'Amazon EMR que vous utilisez.

EMR 5.30.0 and 6.0.0 and later

Example : liste tous les processus en cours

L'exemple suivant utilise `systemctl` et indique à `--type` d'afficher tous les processus.

```
systemctl --type=service
```

Example : Répertorier les processus spécifiques

L'exemple suivant répertorie tous les processus dont les noms contiennent `hadoop`.

```
systemctl --type=service | grep -i hadoop
```

Exemple de sortie :

```
hadoop-hdfs-namenode.service           loaded active running Hadoop namenode
hadoop-httpfs.service                  loaded active running Hadoop httpfs
hadoop-kms.service                     loaded active running Hadoop kms
hadoop-mapreduce-historyserver.service loaded active running Hadoop historyserver
hadoop-state-pusher.service            loaded active running Daemon process that
processes and serves EMR metrics data.
hadoop-yarn-proxyserver.service        loaded active running Hadoop proxyserver
hadoop-yarn-resourcemanager.service    loaded active running Hadoop resourcemanager
hadoop-yarn-timelineserver.service     loaded active running Hadoop timelineserver
```

Example : Afficher un rapport d'état détaillé pour un processus spécifique

L'exemple suivant affiche un rapport d'état détaillé du service `hadoop-hdfs-namenode`.

```
sudo systemctl status hadoop-hdfs-namenode
```

Exemple de sortie :

```
hadoop-hdfs-namenode.service - Hadoop namenode
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:01:46 UTC; 26min ago
 Main PID: 9733 (java)
    Tasks: 0
   Memory: 1.1M
   CGroup: /system.slice/hadoop-hdfs-namenode.service
           # 9733 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server -
XX:OnOutOfMemoryError=kill -9 %p ...

Aug 18 21:01:37 ip-172-31-20-123 systemd[1]: Starting Hadoop namenode...
Aug 18 21:01:37 ip-172-31-20-123 su[9715]: (to hdfs) root on none
Aug 18 21:01:37 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: starting namenode,
 logging to /var/log/hadoop-hdfs/ha...out
Aug 18 21:01:46 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: Started Hadoop
 namenode:[  OK  ]
Aug 18 21:01:46 ip-172-31-20-123 systemd[1]: Started Hadoop namenode.
Hint: Some lines were ellipsized, use -l to show in full.
```

EMR 4.x - 5.29.0

Example : liste tous les processus en cours

L'exemple suivant répertorie tous les processus en cours d'exécution.

```
initctl list
```

EMR 2.x - 3.x

Example : liste tous les processus en cours

L'exemple suivant répertorie tous les processus en cours d'exécution.

```
ls /etc/init.d/
```

Arrêt et redémarrage des processus

Après que vous avez déterminé les processus en cours d'exécution, vous pouvez les arrêter et les redémarrer si nécessaire.

EMR 5.30.0 and 6.0.0 and later

Example : Arrêter un processus

L'exemple suivant arrête le processus `hadoop-hdfs-namenode`.

```
sudo systemctl stop hadoop-hdfs-namenode
```

Vous pouvez interroger le `status` pour vérifier que le processus est arrêté.

```
sudo systemctl status hadoop-hdfs-namenode
```

Exemple de sortie :

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
  Active: failed (Result: exit-code) since Wed 2021-08-18 21:37:50 UTC; 8s ago
Main PID: 9733 (code=exited, status=143)
```

Example : Démarrer un processus

L'exemple suivant lance le processus `hadoop-hdfs-namenode`.

```
sudo systemctl start hadoop-hdfs-namenode
```

Vous pouvez interroger l'état pour vérifier que le processus est en cours d'exécution.

```
sudo systemctl status hadoop-hdfs-namenode
```

Exemple de sortie :

```
hadoop-hdfs-namenode.service - Hadoop namenode
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:38:24 UTC; 2s ago
  Process: 13748 ExecStart=/etc/init.d/hadoop-hdfs-namenode start (code=exited,
 status=0/SUCCESS)
 Main PID: 13800 (java)
    Tasks: 0
```

```
   Memory: 1.1M
   CGroup: /system.slice/hadoop-hdfs-namenode.service
           # 13800 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server
 -XX:OnOutOfMemoryError=kill -9 %p...
```

EMR 4.x - 5.29.0

Example : Arrêter un processus en cours

L'exemple suivant arrête le service `hadoop-hdfs-namenode`.

```
sudo stop hadoop-hdfs-namenode
```

Example : Redémarrer un processus arrêté

L'exemple suivant redémarre le service `hadoop-hdfs-namenode`. Vous devez utiliser la commande `start` et non `restart`.

```
sudo start hadoop-hdfs-namenode
```

Example : Vérifier l'état du processus

Ce qui suit permet de récupérer l'état de `hadoop-hdfs-namenode`. Vous pouvez utiliser la commande `status` pour vérifier que le processus s'est arrêté ou a démarré.

```
sudo status hadoop-hdfs-namenode
```

EMR 2.x - 3.x

Example : Arrêter un processus d'application

L'exemple suivant arrête le service `hadoop-hdfs-namenode`, qui est rattaché à la version d'Amazon EMR installée sur le cluster.

```
sudo /etc/init.d/hadoop-hdfs-namenode stop
```

Example : Redémarrer un processus d'application

L'exemple de commande suivant redémarre le processus `hadoop-hdfs-namenode` :

```
sudo /etc/init.d/hadoop-hdfs-namenode start
```

Example : Arrêter un processus Amazon EMR

L'exemple suivant arrête un processus, tel que le contrôleur d'instance, qui n'est pas rattaché à la version d'Amazon EMR sur le cluster.

```
sudo /sbin/stop instance-controller
```

Example : Redémarrer un processus Amazon EMR

L'exemple suivant redémarre un processus, tel que le contrôleur d'instance, qui n'est pas rattaché à la version d'Amazon EMR sur le cluster.

```
sudo /sbin/start instance-controller
```

> **Note**
>
> Les commandes `/sbin/start, stop` et `restart` sont des liens symboliques vers `/sbin/intictl`. Pour plus d'informations sur `initctl`, consulte la page consacrée à initctl man en entrant `man initctl` à l'invite de commande.

Erreurs courantes dans Amazon EMR

Parfois, les clusters échouent ou mettent du temps à traiter les données. Les sections suivantes répertorient certains problèmes courants liés aux clusters avec des suggestions pour les résoudre.

Rubriques

- Codes d'erreur avec ErrorDetail informations
- Erreurs de ressource
- Erreurs d'entrée et sortie
- Erreurs d'autorisations
- Erreurs de cluster Hive
- Erreurs VPC

- Erreurs de cluster de diffusion en continu
- Erreurs de cluster des fichiers JAR personnalisés
- AWS GovCloud Erreurs (ouest des États-Unis)
- Trouver un cluster manquant

Codes d'erreur avec ErrorDetail informations

Lorsqu'un cluster EMR s'arrête avec une erreur, les API `DescribeCluster` et `ListClusters` renvoient un code d'erreur et un message d'erreur. Pour certaines erreurs de cluster, le tableau de données `ErrorDetail` peut vous aider à résoudre le problème.

Les erreurs qui incluent un tableau `ErrorDetail` fournissent les informations suivantes :

`ErrorCode`

Code d'erreur unique que vous pouvez utiliser pour l'accès par programmation.

`ErrorData`

Liste d'identifiants sous forme de paires clé-valeur que vous pouvez utiliser pour la programmation ou la recherche manuelle. Pour une description des valeurs `ErrorData` incluses dans un code d'erreur, consultez la page de résolution des problèmes rattachée au code d'erreur.

`ErrorMessage`

Description de l'erreur avec un lien vers des informations supplémentaires dans la documentation Amazon EMR.

> **Note**
>
> Nous vous déconseillons d'analyser le texte à partir de `ErrorMessage`, car celui-ci est sujet à modification.

Codes d'erreur par catégorie

- Codes d'erreur de défaillance de l'amorçage
- Codes d'erreur internes
- Codes d'erreur d'échec de validation

Codes d'erreur de défaillance de l'amorçage

Les sections suivantes fournissent des informations de dépannage relatives aux codes d'erreur d'échec de l'amorçage.

Rubriques

- BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE
- BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY
- BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE

Présentation

Lorsqu'un cluster se termine avec une erreur `BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE`, une action d'amorçage a échoué dans l'instance principale. Pour plus d'informations sur les actions d'amorçage, consultez Création d'actions d'amorçage pour installer des logiciels supplémentaires.

Résolution

Pour résoudre cette erreur, passez en revue les informations renvoyées dans l'erreur d'API, modifiez votre script d'action d'amorçage et créez un nouveau cluster avec l'action d'amorçage mise à jour.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`primary-instance-id`

ID de l'instance principale où l'action d'amorçage a échoué.

`bootstrap-action`

Numéro ordinal de l'action d'amorçage qui a échoué. Un script dont la valeur `bootstrap-action` est égale à 1 est la première action d'amorçage exécutée sur l'instance.

`return-code`

Le code de retour de l'action d'amorçage qui a échoué.

`amazon-s3-path`

L'emplacement sur Amazon S3 de l'action d'amorçage qui a échoué.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger la cause première de l'erreur d'action d'amorçage. Lancez ensuite un nouveau cluster.

1. Consultez les fichiers journaux des actions d'amorçage dans Amazon S3 pour identifier la cause première de l'échec. Pour en savoir plus sur la façon de consulter les journaux Amazon EMR, consultez Afficher les fichiers journaux .
2. Si vous avez activé les journaux de cluster lors de la création de l'instance, consultez le journal `stdout`pour plus d'informations. Vous pouvez trouver le journal `stdout` de l'action d'amorçage dans cet emplacement Amazon S3 :

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

   Pour plus d'informations sur les journaux de clusters, consultez Configuration de la journalisation et du débogage du cluster.
3. Pour déterminer l'échec de l'action d'amorçage, passez en revue les exceptions dans les journaux `stdout`et la valeur `return-code` dans `ErrorData`.
4. Utilisez les résultats de l'étape précédente pour revoir votre action d'amorçage afin qu'elle évite les exceptions ou qu'elle puisse gérer les exceptions correctement lorsqu'elles se produisent.
5. Lancez un nouveau cluster avec votre action d'amorçage mise à jour.

BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY

Présentation

Un cluster se termine avec l'erreur `BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY` lorsque l'instance principale ne parvient pas à télécharger un script d'action d'amorçage depuis l'emplacement Amazon S3 que vous spécifiez. Les causes sont généralement les suivantes :

- Le fichier de script d'action d'amorçage ne se trouve pas à l'emplacement Amazon S3 spécifié.
- Le rôle de service pour les instances Amazon EC2 du cluster (également appelé profil d'instance EC2 pour Amazon EMR) n'est pas autorisé à accéder au compartiment Amazon S3 où réside le script d'action d'amorçage. Pour de plus amples informations sur les rôles de service, veuillez consulter Rôle de service pour les instances EC2 de cluster (profil d'instance EC2).

Pour plus d'informations sur les actions d'amorçage, consultez Création d'actions d'amorçage pour installer des logiciels supplémentaires.

Résolution

Pour résoudre cette erreur, assurez-vous que votre instance principale dispose d'un accès approprié au script d'action d'amorçage.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`primary-instance-id`

ID de l'instance principale où l'action d'amorçage a échoué.

`bootstrap-action`

Numéro ordinal de l'action d'amorçage qui a échoué. Un script dont la valeur `bootstrap-action` est égale à 1 est la première action d'amorçage exécutée sur l'instance.

`amazon-s3-path`

L'emplacement sur Amazon S3 de l'action d'amorçage qui a échoué.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger la cause première de l'erreur d'action d'amorçage. Lancez ensuite un nouveau cluster.

Étapes de résolution des problèmes

1. Utilisez la valeur `amazon-s3-path` du tableau `ErrorData` pour trouver le script d'action d'amorçage approprié dans Amazon S3.
2. Si vous avez activé les journaux de cluster lors de la création de l'instance, consultez le journal `stdout`pour plus d'informations. Vous pouvez trouver le journal `stdout` de l'action d'amorçage dans cet emplacement Amazon S3 :

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

   Pour plus d'informations sur les journaux de clusters, consultez Configuration de la journalisation et du débogage du cluster.
3. Pour déterminer l'échec de l'action d'amorçage, passez en revue les exceptions dans les journaux `stdout`et la valeur `return-code` dans `ErrorData`.
4. Utilisez les résultats de l'étape précédente pour revoir votre action d'amorçage afin qu'elle évite les exceptions ou qu'elle puisse gérer les exceptions correctement lorsqu'elles se produisent.
5. Lancez un nouveau cluster avec votre action d'amorçage mise à jour.

BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

Présentation

L'erreur `BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY` indique que l'instance principale ne trouve pas le script d'action d'amorçage qu'elle vient de télécharger depuis le compartiment Amazon S3 spécifié.

Résolution

Pour résoudre cette erreur, assurez-vous que votre instance principale dispose d'un accès approprié au script d'action d'amorçage.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`primary-instance-id`

ID de l'instance principale où l'action d'amorçage a échoué.

`bootstrap-action`

Numéro ordinal de l'action d'amorçage qui a échoué. Un script dont la valeur `bootstrap-action` est égale à 1 est la première action d'amorçage exécutée sur l'instance.

`amazon-s3-path`

L'emplacement sur Amazon S3 de l'action d'amorçage qui a échoué.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger la cause première de l'erreur d'action d'amorçage. Lancez ensuite un nouveau cluster.

1. Pour trouver le script d'action d'amorçage approprié dans Amazon S3, utilisez la valeur `amazon-s3-path` du tableau`ErrorData`.
2. Consultez les fichiers journaux des actions d'amorçage dans Amazon S3 pour identifier la cause première de l'échec. Pour en savoir plus sur la façon de consulter les journaux Amazon EMR, consultez Afficher les fichiers journaux .

   > **Note**
   >
   > Si vous n'avez pas activé les journaux pour votre cluster, vous devez créer un nouveau cluster avec les mêmes configurations et actions d'amorçage. Pour vous assurer que les journaux du cluster sont activés, consultez Configuration de la journalisation et du débogage du cluster.

3. Consultez le journal `stdout` de vos actions d'amorçage et vérifiez qu'aucun processus personnalisé ne supprime les fichiers `/emr/instance-controller/lib/bootstrap-actions` du dossier de vos instances principales. Vous pouvez trouver le journal `stdout` de l'action d'amorçage dans cet emplacement Amazon S3 :

```
s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-
actions/Failed_Bootstrap_Action_Number/stdout.gz
```

4. Lancez un nouveau cluster avec votre action d'amorçage mise à jour.

Codes d'erreur internes

Les sections suivantes fournissent des informations de dépannage pour les codes d'erreur internes.

Rubriques

- INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ
- INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY
- INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ

Présentation

Un cluster se termine avec une erreur `INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ` lorsque la zone de disponibilité sélectionnée ne dispose pas d'une capacité suffisante pour répondre à votre demande de type d'instance Amazon EC2. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité. Pour plus d'informations sur les sous-réseaux pour Amazon EMR, consultez Configuration de la mise en réseau.

Résolution

Pour résoudre cette erreur, modifiez les configurations des types d'instance et créez un nouveau cluster avec votre demande mise à jour.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`instance-type`

Type d'instance dont la capacité est épuisée.

`availability-zone`

Zone de disponibilité vers laquelle correspond votre sous-réseau.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger la cause première de l'erreur de configuration du cluster :

- Consultez les meilleures pratiques pour la configuration d'un cluster. Consultez Bonnes pratiques pour la configuration des clusters dans le Guide de gestion Amazon EMR.
- Résolvez les problèmes de lancement et passez en revue votre configuration. Consultez la section Résolution des problèmes de lancement d'instance dans le guide de l'utilisateur Amazon EC2.
- Lancez un nouveau cluster avec votre configuration de cluster mise à jour.

INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY

Présentation

Un cluster se termine avec une erreur `INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY` lorsqu'Amazon EMR ne peut pas répondre à votre demande d'instance Spot pour le nœud primaire parce que le prix des instances dépassent votre prix Spot maximal. Pour plus d'informations, consultez la section Instances Spot dans le guide de l'utilisateur Amazon EC2.

Résolution

Pour résoudre cette erreur, spécifiez pour votre cluster des types d'instance conformes à votre objectif de prix, ou augmentez votre limite de prix pour le même type d'instance.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`primary-instance-id`

L'ID de l'instance principale du cluster qui a échoué.

`instance-type`

Type d'instance dont la capacité est épuisée.

`availability-zone`

Zone de disponibilité dans laquelle réside votre sous-réseau.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour résoudre les problèmes liés à votre stratégie de configuration de cluster, puis lancez un nouveau cluster :

1. Passez en revue les meilleures pratiques relatives aux instances Spot Amazon EC2 et passez en revue votre stratégie de configuration de cluster. Pour plus d'informations, consultez les meilleures pratiques pour EC2 Spot dans le guide de l'utilisateur Amazon EC2 et. Bonnes pratiques pour la configuration des clusters
2. Modifiez les configurations de votre type d'instance ou votre zone de disponibilité et créez un nouveau cluster avec votre demande mise à jour.
3. Si le problème persiste, utilisez la capacité à la demande pour votre instance principale.

INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

Présentation

Un cluster se termine avec une `INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY` erreur lorsque la capacité est insuffisante pour répondre à une demande d'instance Spot pour votre nœud primaire. Pour plus d'informations, consultez la section Instances Spot dans le guide de l'utilisateur Amazon EC2.

Résolution

Pour résoudre cette erreur, spécifiez pour votre cluster des types d'instance conformes à votre objectif de prix, ou augmentez votre limite de prix pour le même type d'instance.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus

d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`primary-instance-id`

L'ID de l'instance principale du cluster qui a échoué.

`instance-type`

Type d'instance dont la capacité est épuisée.

`availability-zone`

Zone de disponibilité vers laquelle correspond votre sous-réseau.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour résoudre les problèmes liés à votre stratégie de configuration de cluster, puis lancez un nouveau cluster :

1. Passez en revue les meilleures pratiques relatives aux instances Spot Amazon EC2 et passez en revue votre stratégie de configuration de cluster. Pour plus d'informations, consultez les meilleures pratiques pour EC2 Spot dans le guide de l'utilisateur Amazon EC2 et. Bonnes pratiques pour la configuration des clusters
2. Modifiez les configurations de vos types d'instance et créez un nouveau cluster avec votre demande mise à jour.
3. Si le problème persiste, utilisez la capacité à la demande pour votre instance principale.

Codes d'erreur d'échec de validation

Les sections suivantes fournissent des informations de dépannage relatives aux codes d'erreur d'échec de validation.

Rubriques

- VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_INVALID_SSH_KEY_NAME

- VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC

Présentation

Lorsque votre cluster et les sous-réseaux auxquels vous faites référence pour votre cluster appartiennent à des clouds privés virtuels (VPC) différents, le cluster se termine avec une erreur `VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC`. Vous pouvez lancer des clusters avec Amazon EMR avec la configuration des flottes d'instances sur les sous-réseaux d'un VPC. Pour plus d'informations sur les flottes d'instances, consultez Configuration de parcs d'instances dans le Guide de gestion Amazon EMR.

Résolution

Pour résoudre cette erreur, utilisez des sous-réseaux appartenant au même VPC que le cluster.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`vpc`

Pour chaque paire VPC-sous-réseau : l'ID du VPC auquel appartient le sous-réseau.

`subnet`

Pour chaque paire VPC-sous-réseau : l'ID du sous-réseau.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger l'erreur :

1. Passez en revue les ID de sous-réseau répertoriés dans le tableau `ErrorData` et confirmez qu'ils appartiennent au VPC sur lequel vous souhaitez lancer le cluster EMR.
2. Modifiez les configurations de vos sous-réseaux. Vous pouvez utiliser l'une des méthodes suivantes pour rechercher tous les sous-réseaux publics et privés disponibles dans un VPC.

- Accédez à la console Amazon VPC. Choisissez Subnets et listez tous les sous-réseaux qui résident au sein Région AWS de votre cluster. Pour rechercher uniquement les sous-réseaux publics ou privés, appliquez le filtre d'attribution automatique d'adresses IPv4 publiques. Pour rechercher et sélectionner des sous-réseaux dans le VPC utilisé par votre cluster, utilisez l'option Filtrer par VPC. Pour plus d'informations sur la création de sous-réseaux, consultez la section Créer un sous-réseau du Guide de l'utilisateur du cloud privé virtuel Amazon.
- Utilisez le AWS CLI pour rechercher tous les sous-réseaux publics et privés disponibles dans le VPC utilisé par votre cluster. Pour plus d'informations, consultez l'API describe-subnets. Pour créer de nouveaux sous-réseaux dans un VPC, consultez l'API create-subnet.

3. Lancez un nouveau cluster avec des sous-réseaux provenant du même VPC que le cluster.

VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC

Présentation

Lorsque votre cluster et les groupes de sécurité que vous lui attribuez appartiennent à des clouds privés virtuels (VPC) différents, le cluster se termine avec une erreur `VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC`. Pour de plus amples informations sur les groupes de sécurité consultez Spécification des groupes de sécurité gérés par Amazon EMR et des groupes de sécurité supplémentaires et Contrôle du trafic réseau avec des groupes de sécurité.

Résolution

Pour résoudre cette erreur, utilisez des groupes de sécurité appartenant au même VPC que le cluster.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`vpc`

Pour chaque paire groupe de sécurité-VPC, l'ID du VPC auquel appartient le groupe de sécurité.

`security-group`

Pour chaque paire groupe de sécurité-VPC, l'ID du groupe de sécurité.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger l'erreur :

1. Passez en revue les ID des groupes de sécurité répertoriés dans le tableau `ErrorData` et confirmez qu'ils appartiennent au VPC sur lequel vous souhaitez lancer le cluster EMR.
2. Accédez à la console Amazon VPC. Choisissez Groupes de sécurité pour répertorier tous les groupes de sécurité de la région que vous sélectionnez. Recherchez les groupes de sécurité du même VPC que votre cluster, puis modifiez la configuration de votre groupe de sécurité.
3. Lancez un nouveau cluster avec des groupes de sécurité issus du même VPC que le cluster.

VALIDATION_ERROR_INVALID_SSH_KEY_NAME

Présentation

Un cluster se termine avec une erreur `VALIDATION_ERROR_INVALID_SSH_KEY_NAME` lorsque vous utilisez une paire de clés Amazon EC2 qui n'est pas valide pour accéder à l'instance principale par SSH. Le nom de la paire de clés est peut-être incorrect ou la paire de clés n'existe peut-être pas dans le fichier demandé Région AWS. Pour plus d'informations sur les paires de clés, consultez les paires de clés Amazon EC2 et les instances Linux dans le guide de l'utilisateur Amazon EC2.

Résolution

Pour résoudre cette erreur, créez un nouveau cluster avec un nom de paire de clés SSH valide.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`ssh-key`

Le nom de la paire de clés SSH que vous avez fourni lors de la création du cluster.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger l'erreur :

1. Vérifiez le fichier *keypair*.pem et assurez-vous qu'il correspond au nom de la clé SSH qui apparaît dans la console Amazon EMR.
2. Accédez à la console Amazon EC2. Vérifiez que le nom de clé SSH que vous avez utilisé est disponible dans Région AWS celui utilisé par votre cluster. Vous trouverez votre numéro de compte à Région AWS côté de votre numéro de compte en haut du AWS Management Console.
3. Lancez un nouveau cluster avec un nom de clé SSH valide.

VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

Présentation

Un cluster se termine avec une erreur `VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED` lorsque les zones de disponibilité Région AWS de votre cluster ne prennent pas en charge le type d'instance spécifié pour un ou plusieurs groupes d'instances. Amazon EMR peut prendre en charge un type d'instance dans une zone de disponibilité au sein d'une région, mais pas dans une autre. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité au sein de la région. Pour consulter la liste des types d'instance et des régions pris en charge par Amazon EMR, consultez Types d’instance pris en charge.

Résolution

Pour résoudre cette erreur, spécifiez les types d'instances pour votre cluster pris en charge par Amazon EMR dans la région et la zone de disponibilité où vous demandez le cluster.

Pour résoudre les problèmes liés au cluster EMR défaillant, reportez-vous aux informations `ErrorDetail` renvoyées par `DescribeCluster` et les API `ListClusters`. Pour plus d’informations, consultez Codes d'erreur avec ErrorDetail informations. Le tableau `ErrorData` dans `ErrorDetail` renvoie les informations suivantes pour ce code d'erreur :

`instance-types`

La liste des types d'instances non pris en charge.

`availability-zones`

La liste des zones de disponibilité vers lesquelles votre sous-réseau est résolu.

`public-doc`

URL publique de la documentation du code d'erreur.

Étapes à suivre

Procédez comme suit pour identifier et corriger l'erreur :

1. Utilisez le AWS CLI pour récupérer les types d'instances disponibles dans une zone de disponibilité. Pour ce faire, vous pouvez utiliser la `ec2 describe-instance-type-offerings` commande pour filtrer les types d'instances disponibles par emplacement (Région AWS ou zone de disponibilité). Par exemple, la commande suivante renvoie les types d'instances proposés dans la zone de disponibilité spécifié, *`us-east-2a`*.

   ```
   aws ec2 describe-instance-type-offerings --location-type "availability-
   zone" --filters Name=location,Values=us-east-2a --region us-east-2 --query
    "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
   ```

   Pour plus d'informations sur la manière de découvrir les types d'instance disponibles, consultez Rechercher un type d'instance Amazon EC2.
2. Après avoir déterminé les types d'instances disponibles dans la même région et zone de disponibilité que le cluster, choisissez l'une des solutions suivantes pour continuer :

   a. Créez un nouveau cluster et choisissez un sous-réseau pour le cluster qui se trouve dans une zone de disponibilité où le type d'instance que vous avez sélectionné est disponible et pris en charge par Amazon EMR.

   b. Créez un nouveau cluster dans la même région et dans le même sous-réseau Amazon EC2 que le cluster défaillant, mais avec un type d'instance pris en charge à cet emplacement par Amazon EMR.

Pour consulter la liste des types d'instance et des régions pris en charge par Amazon EMR, consultez Types d'instance pris en charge. Pour comparer les capacités des types d'instance, consultez les types d'instance Amazon EC2.

Erreurs de ressource

Les erreurs suivantes sont généralement causées par des ressources limitées sur le cluster.

Rubriques

- Le cluster se termine avec NO_SLAVE_LEFT et les nœuds principaux FAILED_BY_MASTER.
- Cannot replicate block, only managed to replicate to zero nodes (Impossible de répliquer un bloc, réplication sur zéro nœud gérée uniquement)
- EC2 QUOTA EXCEEDED (QUOTA EC2 DÉPASSÉ)
- Too many fetch-failures (Trop d'erreurs de récupération)
- Le fichier a pu uniquement être répliqué sur 0 nœud et non 1
- Nœuds figurant sur la liste des refus
- Erreur de Limitation
- Type d'instance non pris en charge
- EC2 est en manque de capacité

Le cluster se termine avec NO_SLAVE_LEFT et les 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 pour un nœud principal dépasse le seuil d'utilisation, le service de NodeManager santé YARN signale le nœud comme`UNHEALTHY`. Lorsqu'il est dans cet état, Amazon EMR met le nœud sur la liste noire et n'y alloue pas de conteneurs YARN. Si le nœud reste non sain pendant 45 minutes, Amazon EMR marque l'instance Amazon EC2 rattachée pour la terminaison en tant que `FAILED_BY_MASTER`. Lorsque toutes les instances Amazon EC2 rattachées à des nœuds principaux sont marquées pour la résiliation, le cluster se résilie avec l'état `NO_SLAVE_LEFT`, car il n'y a pas de ressources pour exécuter des 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. 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 Utiliser la mise à l'échelle des clusters.

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 rattachée 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 Utilisation de la protection contre la résiliation.

Créer une alarme pour la UnhealthyNodes CloudWatch métrique MR

Cette métrique indique le nombre de nœuds de rapports d'un état `UNHEALTHY`. Elle est équivalente à la métrique YARN `mapred.resourcemanager.NoOfUnhealthyNodes`. Vous pouvez configurer une notification pour cette alarme afin de vous avertir des nœuds qui ne sont pas sains avant que le délai d'attente de 45 minutes soit atteint. Pour plus d'informations, consultez Surveillance des métriques Amazon EMR avec CloudWatch.

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 de plus amples informations, veuillez consulter Configuration des applications dans le guide de version Amazon EMR. Vous pouvez également vous connecter aux instances Amazon EC2 rattachées 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

effectué la modification, vous devez redémarrer hadoop-yarn-nodemanager comme indiqué ci-dessous.

> **Important**
>
> Lorsque vous redémarrez le NodeManager service, les conteneurs YARN actifs sont détruits sauf `yarn.nodemanager.recovery.enabled` s'ils sont configurés pour `true` utiliser la classification de `yarn-site` configuration lors de la création du 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	Valeur par défaut	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	Fraction minimale du nombre de disques qui doivent être sains pour NodeManager lancer 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-`

Propriété	Valeur par défaut	Description
		`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, NodeManager le disque est plein. 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)

Erreur : « Impossible de répliquer un bloc, réplication sur zéro nœud gérée uniquement » se produit généralement lorsqu'un cluster ne dispose pas d'un espace de stockage HDFS suffisant. 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 utilisées 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 plus d'informations sur la quantité de stockage local pour chaque type d'instance EC2, consultez la section Types et familles d'instances dans le guide de l'utilisateur Amazon EC2.

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 aurait 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 peuvent 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 :

- Suivez les instructions figurant dans AWS Service Quotas dans la Référence générale d'Amazon Web Services pour demander une augmentation de la limite de service. Pour certaines API, il peut être préférable de configurer un CloudWatch événement plutôt que d'augmenter les limites. Pour en savoir plus, consultez Quand configurer des événements EMR dans CloudWatch.
- 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 cartographique reste à 100 % dans les journaux, examinez les

CloudWatch indicateurs, en particulier`RunningMapTasks`, pour vérifier si la tâche cartographique 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 Création d'actions d'amorçage pour installer des logiciels supplémentaires.
- 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.
- 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 cette erreur s'affiche, cela signifie que le NameNode démon ne dispose d'aucune DataNode instance disponible sur laquelle 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.
- DataNode les instances n'étaient peut-être pas disponibles lors de l'exécution de la tâche.

- DataNode les instances peuvent avoir été empêchées de communiquer 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 JobTracker démon n'est peut-être pas autorisé à créer des informations de suivi des tâches.
- Le paramètre d'espace réservé pour une DataNode instance 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 au manque d'espace disque de HDFS, examinez la `HDFSUtilization` métrique contenue dans CloudWatch. Si cette valeur est trop élevée, vous pouvez ajouter des nœuds principaux supplémentaires au cluster. Si vous pensez que votre cluster risque de manquer d'espace disque HDFS, vous pouvez configurer une alarme CloudWatch pour vous avertir lorsque la valeur de `HDFSUtilization` dépasse un certain niveau. Pour plus d'informations, consultez Redimensionnement manuel d'un cluster en cours d'exécution et Surveillance des métriques Amazon EMR avec CloudWatch.

Si le problème n'est pas dû au manque d'espace du HDFS, vérifiez les DataNode journaux, les NameNode journaux et la connectivité réseau pour détecter d'autres problèmes qui auraient pu empêcher HDFS de répliquer les données. Pour plus d'informations, consultez Afficher les fichiers journaux .

Nœuds figurant sur la liste des refus

Le NodeManager daemon est responsable du lancement et de la gestion des conteneurs sur les nœuds principaux et les nœuds de tâches. Les conteneurs sont alloués au NodeManager daemon par le ResourceManager daemon qui s'exécute sur le nœud maître. Le ResourceManager surveille le NodeManager nœud par un battement de cœur.

Dans certaines situations, le ResourceManager daemon deny répertorie a NodeManager, le supprimant du pool de nœuds disponibles pour traiter les tâches :

- Si aucun battement de cœur n' NodeManager a été envoyé au ResourceManager daemon au cours des 10 dernières minutes (600 000 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 Guide de version Amazon EMR.
- NodeManager vérifie l'état des disques déterminé par `yarn.nodemanager.local-dirs` et`yarn.nodemanager.log-dirs`. Les vérifications incluent les autorisations et l'espace disque

disponible (< 90 %). Si un disque échoue à la vérification, il NodeManager cesse de l'utiliser mais indique toujours que l'état du nœud est sain. Si plusieurs disques échouent à la vérification, le nœud est signalé comme étant défectueux ResourceManager et aucun nouveau conteneur ne lui est attribué.

Le responsable de l'application peut également refuser de NodeManager répertorier un nœud si plus de trois tâches ont échoué. Vous pouvez le remplacer par une valeur plus élevée à l'aide du paramètre de configuration `mapreduce.job.maxtaskfailures.per.tracker`. D'autres paramètres de configuration que vous pouvez modifier contrôlent le nombre de tentatives pour une tâche avant de l'indiquer comme ayant échoué : `mapreduce.map.max.attempts` pour les tâches Map et `mapreduce.reduce.maxattempts` pour les tâches Reduce. Pour plus d'informations sur la modification des paramètres de configuration, consultez Configuration des applications dans le Guide de version Amazon EMR.

Erreur de Limitation

Les erreurs « Limité par *Amazon EC2* lors du lancement du cluster » et « Impossible de mettre en service les instances en raison d'une limitation de *Amazon EC2* » se produisent lorsqu'Amazon EMR ne peut pas traiter une demande parce qu'un autre service a limité l'activité. Amazon EC2 est la source la plus courante d'erreurs de régulation, mais d'autres services peuvent être à l'origine d'erreurs de régulation. Les limites de service AWS s'appliquent par région afin d'améliorer les performances, et une erreur de régulation indique que vous avez dépassé la limite de service de 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.
- 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.
- 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.

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 gère les demandes d'API, consultez Interroger le débit de demandes d'API dans Référence d'API Amazon EC2.

Solutions

Envisagez les solutions suivantes :

- Suivez les instructions figurant dans AWS Service Quotas dans la Référence générale d'Amazon Web Services pour demander une augmentation de la limite de service. Pour certaines API, il peut être préférable de configurer un CloudWatch événement plutôt que d'augmenter les limites. Pour en savoir plus, consultez Quand configurer des événements EMR dans CloudWatch.
- Si vous avez des clusters qui se lancent au même moment (par exemple, en début d'heure), envisagez d'échelonner les heures de démarrage.
- 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 de la mise à l'échelle automatique avec une politique personnalisée pour les groupes d'instances.

Type d'instance non pris en charge

Si vous créez un cluster et que celui-ci échoue avec le message d'erreur « Le type d'instance demandé n'*`InstanceType`*est pas pris en charge dans la zone de disponibilité demandée », cela signifie que vous avez créé le cluster et que vous avez spécifié un type d'instance pour un ou plusieurs groupes d'instances qui n'est pas pris en charge par Amazon EMR dans la région et la zone de disponibilité où le cluster a été créé. Amazon EMR peut prendre en charge un type d'instance dans une zone de disponibilité d'une région et pas dans une autre. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité au sein de la région.

Solution

Déterminez 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`* par le type d'instance que vous souhaitez utiliser,

ami-035be7bafff33b6b6 par l'AMI rattachée à ce type d'instance, et *subnet-12ab3c45* par 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
```

Pour obtenir des instructions sur la recherche d'un ID d'AMI, consultez Trouver une AMI Linux. Pour trouver un ID de sous-réseau, vous pouvez utiliser la commande describe-subnets.

Pour plus d'informations sur la manière de découvrir les types d'instance disponibles, consultez Rechercher un type d'instance Amazon EC2.

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’instance pris en charge. Pour comparer les capacités des types d'instance EC2, consultez les 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 est en manque de capacité

Une erreur « EC2 est en panne de capacité *InstanceType*» se produit lorsque vous tentez de créer un cluster, ou d'ajouter des instances à un cluster, dans une zone de disponibilité qui ne contient plus le type d'instance EC2 spécifié. Le sous-réseau que vous sélectionnez pour un cluster détermine la zone de disponibilité.

Pour créer un cluster, procédez comme suit :

- Spécifiez un autre type d'instance doté de fonctionnalités similaires
- Créez le cluster dans une autre région
- Sélectionnez un sous-réseau dans une zone de disponibilité où le type d'instance souhaité peut être disponible.

Pour ajouter des instances à un cluster en cours d'exécution, effectuez l'une des opérations suivantes :

- Modifiez des configurations de groupe d'instances ou des configurations de flotte d'instances pour ajouter des types d'instance disponibles avec des capacités similaires. Pour obtenir la liste des types d’instances, consultez Types d’instance pris en charge. Pour comparer les capacités des types d'instance EC2, consultez les types d'instance Amazon EC2.
- Arrêtez le cluster et recréez-le dans une région et une zone de disponibilité dans lesquelles le type d'instance est disponible.

Erreurs d'entrée et sortie

Les erreurs suivantes sont courantes dans les opérations d'entrée et sortie de cluster.

Rubriques

- Votre chemin d'accès à Amazon Simple Storage Service (Amazon S3) comporte-t-il au moins trois barres obliques ?
- Essayez-vous de parcourir de façon récursive les répertoires d'entrée ?
- Votre répertoire de sortie existe-t-il déjà ?
- Essayez-vous de spécifier une ressource à l'aide d'une URL HTTP ?
- Référencez-vous un compartiment Amazon S3 à l'aide d'un format de nom non valide ?
- Rencontrez-vous des difficultés lors du chargement des données vers ou depuis Amazon S3 ?

Votre chemin d'accès à Amazon Simple Storage Service (Amazon S3) comporte-t-il au moins trois barres obliques ?

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 comme « s3n://DOC-EXAMPLE-BUCKET1 », vous devriez utiliser « s3n://DOC-EXAMPLE-BUCKET1/ », sinon Hadoop fait échouer votre cluster dans la plupart des cas.

Essayez-vous de parcourir de façon récursive les répertoires d'entrée ?

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.

Votre répertoire de sortie existe-t-il déjà ?

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.

Essayez-vous de spécifier une ressource à l'aide d'une URL HTTP ?

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.

Référencez-vous un compartiment Amazon S3 à l'aide d'un format de nom non valide ?

Si vous tentez d'utiliser un nom de compartiment tel que « DOC-EXAMPLE-BUCKET1.1 » avec Amazon EMR, votre cluster échouera car Amazon EMR exige que les noms de compartiment soient des noms d'hôte 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 plus d'informations sur la manière de formater les noms des compartiments Amazon S3, consultez la section Restrictions et limitations des compartiments du Guide de l'utilisateur Amazon Simple Storage Service.

Rencontrez-vous des difficultés lors du chargement des données vers ou depuis 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.

- Si une erreur interne se produit dans Amazon S3, votre application doit gérer cela de façon appropriée et répéter l'opération.
- Si des appels à Amazon S3 mettent trop longtemps à être retournés, votre application peut avoir besoin de réduire la fréquence à laquelle elle appelle Amazon S3.
- Répertorier tous les objets figurant dans un compartiment Amazon S3 représente un appel onéreux. Votre application doit réduire au maximum ce nombre d'opérations.

Il existe plusieurs façons d'améliorer la façon dont votre cluster interagit avec Amazon S3.

- Lancez votre cluster à l'aide de la dernière version d'Amazon EMR.
- Utilisez S3 DistCp pour déplacer des objets dans et hors d'Amazon S3. S3 DistCp implémente la gestion des erreurs, les nouvelles tentatives et les interruptions pour répondre aux exigences d'Amazon S3. Pour plus d'informations, consultez la section Copie distribuée à l'aide de S3 DistCp.
- Concevez votre application en gardant à l'esprit la cohérence à terme. Utilisez le système HDFS pour le stockage de données intermédiaires quand le cluster est en cours d'exécution et Amazon S3 uniquement pour entrer les données initiales et générer les résultats finaux.
- Si vos clusters valident 200 transactions ou plus par seconde sur Amazon S3, contactez l'assistance pour préparer votre compartiment à un plus grand nombre de transactions par seconde et envisagez d'utiliser les stratégies de partition clés décrites dans Conseils et astuces sur les performances d'Amazon S3.
- Attribuez au paramètre de configuration Hadoop io.file.buffer.size la valeur 65536. Hadoop passe ainsi moins de temps à chercher dans les objets Amazon S3.
- Envisagez de désactiver la fonctionnalité d'exécution spéculative de Hadoop si votre cluster connaît des problèmes de simultanéité avec Amazon S3. Cela s'avère utile également pour dépanner un cluster lent. Pour ce faire, définissez les propriétés `mapreduce.map.speculative` et `mapreduce.reduce.speculative` sur `false`. Lorsque vous lancez un cluster, vous pouvez définir ces valeurs l'aide de la classification de configuration `mapred-env`. Pour de plus amples informations, veuillez consulter Configuration des applications dans le guide de version Amazon EMR.
- Si vous exécutez un cluster Hive, consultez Rencontrez-vous des problèmes de chargement de données vers ou depuis Amazon S3 dans Hive ?.

Pour plus d'informations, consultez les meilleures pratiques relatives aux erreurs Amazon S3 dans le Guide de l'utilisateur d'Amazon Simple Storage Service.

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 ?
- Si vous utilisez IAM, les politiques Amazon EC2 appropriées sont-elles définies ?

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 primaire à l'aide de SSH.

Si vous utilisez IAM, les politiques Amazon EC2 appropriées sont-elles définies ?

Étant donné qu'Amazon EMR utilise des instances EC2 en tant que nœuds, les utilisateurs Amazon EMR ont également besoin de disposer de certaines politiques Amazon EC2 afin qu'Amazon EMR puisse gérer ces instances pour l'utilisateur. Si les autorisations requises ne sont pas définies, Amazon EMR renvoie l'erreur : « Compte non autorisé à appeler EC2. »

Pour de plus amples informations sur les politiques Amazon EC2 que votre compte IAM doit définir pour exécuter Amazon EMR, consultez Fonctionnement d'Amazon EMR avec IAM.

Erreurs de cluster Hive

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

- Utilisez-vous la dernière version de Hive ?
- Avez-vous rencontré une erreur de syntaxe dans le script Hive ?
- Une tâche a-t-elle échoué lors d'une exécution interactive ?
- Rencontrez-vous des problèmes de chargement de données vers ou depuis Amazon S3 dans Hive ?

Utilisez-vous la dernière version de Hive ?

La dernière version de Hive comporte tous les correctifs et correctifs de bogues actuels, ce qui peut résoudre votre problème.

Avez-vous rencontré une erreur de syntaxe dans le script Hive ?

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 Afficher les fichiers journaux .

Une tâche a-t-elle échoué lors d'une exécution interactive ?

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 Afficher les fichiers journaux .

Rencontrez-vous des problèmes de chargement de données vers ou depuis Amazon S3 dans Hive ?

Si vous rencontrez des difficultés pour accéder aux données dans Amazon S3, commencez par vérifier les causes possibles répertoriées dans Rencontrez-vous des difficultés lors du chargement des données vers ou depuis Amazon S3 ?. Si aucun de ces problèmes n'est à l'origine, vous pouvez utiliser les options spécifiques à Hive suivantes.

- Veillez à utiliser la dernière version de Hive qui comporte tous les correctifs et correctifs de bogues actuels qui peuvent résoudre votre problème. Pour plus d'informations, consultez Apache Hive.
- L'utilisation de `INSERT OVERWRITE` nécessite l'affichage du contenu du compartiment ou du dossier Amazon S3. Il s'agit d'une opération coûteuse. Si possible, réduisez manuellement le chemin d'accès plutôt que de faire répertorier et supprimer des objets existants par Hive.
- Si vous utilisez une version antérieure à la version 5.0 d'Amazon EMR, vous pouvez utiliser la commande suivante dans HiveQL afin de mettre en pré-cache les résultats d'une opération de liste Amazon S3 localement sur le cluster :

```
set hive.optimize.s3.query=true;
```

- Si possible, utilisez des partitions statiques.

- Dans certaines versions de Hive et d'Amazon EMR, il est possible que l'utilisation de ALTER TABLES échoue, car le tableau est stockée dans un autre emplacement que celui prévu par Hive. La solution consiste à ajouter ou mettre à jour les éléments suivants dans `/home/hadoop/conf/core-site.xml`:

```
<property>
    <name>fs.s3n.endpoint</name>
    <value>s3.amazonaws.com</value>
</property>
```

Erreurs VPC

Les erreurs suivantes sont communes à la configuration de VPC dans Amazon EMR.

Rubriques

- Configuration de sous-réseau non valide
- Jeu d'options DHCP manquant
- Erreurs d'autorisations
- Erreurs qui se traduisent par START_FAILED
- Cluster Terminated with errors et NameNode ne parvient pas à démarrer

Configuration de sous-réseau non valide

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.

Jeu d'options DHCP manquant

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.
```

or

```
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 (USA Ouest), définissez domain-name sur au **`us-gov-west-1.compute.internal`** lieu de la valeur utilisée dans l'exemple suivant.

- 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`**. Par exemple, dans la région us-west-2, utilisez domain-name (nom-domaine)=**`us-west-2.compute.internal`**.
- domain-name-servers (serveurs-nom-domaine) = **`AmazonProvidedDNS`**

Pour plus d'informations, consultez Jeux d'options DHCP.

Erreurs d'autorisations

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 le ActionOnFailure paramètre est défini surTERMINATE_JOB_FLOW, le cluster se terminera avec l'état,SHUTDOWN_COMPLETED_WITH_ERRORS.

Quelques façons pour résoudre ce problème incluent les suivantes :

- Si vous utilisez une politique de compartiment Amazon S3 au sein d'un VPC, assurez-vous de donner accès à tous les compartiments en créant un point de terminaison de VPC et en sélectionnant Tout autoriser sous l'option Politique lorsque vous créez le point de terminaison.
- Assurez-vous que toutes stratégies rattachées aux ressources S3 incluent le VPC dans lequel vous lancez le cluster.
- Essayez d'exécuter la commande suivante à partir de votre cluster afin de vérifier que vous pouvez accéder au compartiment

  ```
  hadoop fs -copyToLocal s3://path-to-bucket /tmp/
  ```

- Vous pouvez obtenir des informations de débogage plus spécifiques en définissant le paramètre `log4j.logger.org.apache.http.wire` sur `DEBUG` dans le fichier `/home/hadoop/conf/log4j.properties` sur le cluster. Vous pouvez vérifier le fichier journal `stderr` après avoir essayé d'accéder au compartiment depuis le cluster. Le fichier journal fournira des informations plus détaillées :

  ```
  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]"
  ```

Erreurs qui se traduisent par **START_FAILED**

Avant l'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 personnalisé 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 mappé par

Amazon EMR qui en résulte serait `ip-10.0.1.0.customdomain.com`. Une entrée est ajoutée dans `/etc/hosts` pour résoudre le nom d'hôte sur 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 **`Terminated with errors`** et NameNode ne parvient pas à démarrer

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 dû à l' NameNode impossibilité de démarrer. Cela entraînera la découverte de l'erreur suivante dans les NameNode journaux, dont l'URI Amazon S3 est de la forme `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.

Erreurs de cluster de diffusion en continu

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

- Les données sont-elles envoyées au mappeur dans un format incorrect ?
- Votre script arrive-t-il à expiration ?
- Transmettez-vous des arguments de diffusion en continu non valides ?
- Votre script s'est-il terminé par une erreur ?

Les données sont-elles envoyées au mappeur dans un format incorrect ?

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 Afficher les fichiers journaux .

Votre script arrive-t-il à expiration ?

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 Afficher les fichiers journaux .

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`.

Transmettez-vous des arguments de diffusion en continu non valides ?

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.

Votre script s'est-il terminé par une erreur ?

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 Afficher les fichiers journaux .

Erreurs de cluster des fichiers JAR personnalisés

Les erreurs suivantes sont communes aux clusters des fichiers JAR personnalisés.

Rubriques

- Votre fichier JAR lève-t-il une exception avant de créer un travail ?
- Votre fichier JAR génère-t-il une erreur au sein d'une tâche de mappage ?

Votre fichier JAR lève-t-il une exception avant de créer un travail ?

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 Afficher les fichiers journaux .

Votre fichier JAR génère-t-il une erreur au sein d'une tâche de mappage ?

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 Afficher les fichiers journaux .

AWS GovCloud Erreurs (ouest des États-Unis)

La région AWS GovCloud (USA Ouest) se distingue des autres régions en termes de sécurité, de configuration et de paramètres par défaut. Par conséquent, utilisez la liste de contrôle suivante pour résoudre les erreurs Amazon EMR spécifiques à la région AWS GovCloud (ouest des États-Unis) avant de suivre des recommandations de dépannage plus générales.

- Vérifiez que vos rôles IAM sont correctement configurés. Pour plus d'informations, consultez Configuration des rôles de service IAM pour les autorisations Amazon EMR aux services et ressources AWS ..
- Assurez-vous que votre configuration de VPC a configuré correctement la prise en charge de nom d'hôte/résolution DNS, la passerelle Internet et les paramètres de jeu d'options DHCP. Pour plus d'informations, consultez Erreurs VPC.

Si ces étapes ne corrigent pas le problème, poursuivez avec les étapes de dépannage des erreurs courantes Amazon EMR. Pour plus d'informations, consultez Erreurs courantes dans Amazon EMR.

Trouver un cluster manquant

Si votre cluster ne figure pas dans la liste des consoles ou dans l'API `ListClusters`, vérifiez les points suivants :

- Confirmez que l'âge du cluster à partir de la date d'achèvement est inférieur à deux mois. Amazon EMR conserve gratuitement pendant deux mois les informations relatives aux métadonnées des clusters terminés. Vous ne pouvez pas supprimer les clusters terminés de la console. Au lieu de cela, Amazon EMR purge automatiquement les clusters terminés au bout de deux mois.
- Confirmez que vous disposez des autorisations de rôle pour afficher le cluster.
- Vérifiez que vous visualisez le même Région AWS endroit où réside le cluster.

Dépannage d'un cluster ayant échoué

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.

> **Note**
>
> Lorsqu'un cluster EMR s'arrête avec une erreur, les API `DescribeCluster` et `ListClusters` renvoient un code d'erreur et un message d'erreur. Pour certaines erreurs de cluster, le tableau de données `ErrorDetail` peut également vous aider à résoudre le problème. Pour plus d'informations, consultez Codes d'erreur avec ErrorDetail informations.

Si votre cluster s'exécute mais met du temps à renvoyer des résultats, consultez Résolution des problèmes de rapidité d'un cluster.

Rubriques

- Étape 1 : Rassembler des données sur le problème
- Étape 2 : Vérifier l'environnement
- Étape 3 : Examiner le dernier changement d'état

- Étape 4 : Examiner les fichiers journaux
- Étape 5 : Test du cluster étape par étape

Étape 1 : Rassembler des données sur le problème

La première étape du dépannage d'un cluster consiste à recueillir des informations sur le problème rencontré, ainsi que sur l'état actuel et la configuration du cluster. Ces informations seront utilisées dans les étapes suivantes pour confirmer ou exclure les causes possibles du problème.

Définition du problème

Définir clairement le problème est la première étape de résolution. Quelques questions à vous poser :

- Quel était l'effet attendu ? Que s'est-il passé à la place ?
- Quand ce problème s'est-il produit pour la première fois ? Combien de fois cela s'est-il produit depuis ?
- Est-ce que quelque chose a changé dans la façon dont je configure ou gère mon cluster ?

Détails du cluster

Les détails du cluster suivants sont utiles pour détecter les problèmes. Pour plus d'informations sur la manière de spécifier ces informations, consultez Afficher l'état et les détails d'un cluster.

- Identifiant du cluster. (Également appelé identifiant de flux de travail.)
- Région AWS et la zone de disponibilité dans laquelle le cluster a été lancé.
- État du cluster, y compris les détails du dernier changement d'état.
- Type et nombre d'instances EC2 spécifiées pour les nœuds principaux et de tâche.

Étape 2 : Vérifier l'environnement

Amazon EMR opère dans le cadre d'un écosystème de services Web et de logiciels open source. Des éléments affectant ces dépendances peuvent attribuer les performances d'Amazon EMR.

Rubriques

- Recherche d'interruptions de service
- Recherche des limites d'utilisation

- Vérification de la version
- Vérifiez la configuration du sous-réseau Amazon VPC

Recherche d'interruptions de service

Amazon EMR utilise plusieurs services Web Amazon en interne. Il exécute des serveurs virtuels sur Amazon EC2, stocke des données et des scripts sur Amazon S3 et transmet des métriques à. CloudWatch Les événements qui perturbent ces services sont rares, mais lorsqu'ils se produisent, ils peuvent entraîner des problèmes dans Amazon EMR.

Avant d'aller plus loin, consultez le Tableau de bord de l'état des services. Vérifiez la région dans laquelle vous avez lancé votre cluster pour voir s'il y a des interruptions dans l'un de ces services.

Recherche des limites d'utilisation

Si vous lancez un cluster de grande taille, si vous avez lancé plusieurs clusters simultanément ou si vous êtes un utilisateur partageant un cluster Compte AWS avec d'autres utilisateurs, le cluster a peut-être échoué car vous avez dépassé une limite de AWS service.

Amazon EC2 limite le nombre d'instances de serveurs virtuels exécutées dans une même AWS région à 20 instances réservées ou à la demande. Si vous lancez un cluster comportant plus de 20 nœuds, ou si vous lancez un cluster dont le nombre total d'instances EC2 actives dépasse 20, le cluster ne sera pas en mesure de lancer toutes les instances EC2 dont il a besoin et risque d'échouer. Compte AWS Dans ce cas, Amazon EMR renvoie une erreur `EC2 QUOTA EXCEEDED`. Vous pouvez demander à AWS augmenter le nombre d'instances EC2 que vous pouvez exécuter sur votre compte en soumettant une demande d'augmentation de la limite d'instance Amazon EC2.

Le délai entre la fermeture d'un cluster et le moment où il libère toutes ses ressources peut également vous faire dépasser vos limites d'utilisation. Selon sa configuration, il peut s'écouler entre 5 et 20 minutes avant que le cluster ne se termine complètement et ne libère les ressources qui lui ont été allouées. 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. 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 nouveau compartiment qui dépasse cette limite, la création du compartiment échouera et peut entraîner l'échec du cluster.

Vérification de la 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.

Vérifiez la configuration du sous-réseau Amazon VPC

Si votre cluster a été lancé dans un sous-réseau Amazon VPC, celui-ci doit être configuré comme décrit dans Configuration de la mise en réseau. Vérifiez également que le sous-réseau dans lequel vous lancez le cluster possède suffisamment d'adresses IP élastiques libres pour en attribuer une à chaque nœud du cluster.

Étape 3 : Examiner le dernier changement d'état

Le dernier changement d'état fournit des informations sur ce qui s'est passé la dernière fois que le cluster a changé d'état. Il contient souvent des informations explicitant les problèmes qui se sont posés lorsque l'état d'un cluster change et devient `FAILED`. Par exemple, si vous lancez un cluster de streaming 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 de streaming 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 Afficher l'état et les détails d'un cluster.

Étape 4 : Examiner les fichiers journaux

L'étape suivante consiste à examiner les fichiers journaux afin de trouver un code d'erreur ou une autre indication du problème rencontré par votre cluster. Pour plus d'informations sur les fichiers journaux disponibles, où les trouver et comment les consulter, consultez Afficher les fichiers journaux .

Il faudra peut-être effectuer un certain travail d'enquête pour déterminer ce qui s'est passé. Hadoop exécute le travail des tâches lors de tentatives de tâches sur différents nœuds du cluster. Amazon EMR peut lancer des tentatives de tâches spéculatives, mettant fin aux autres tentatives de tâches qui n'aboutissent pas. Cela génère une activité importante qui est enregistrée au fur et à

mesure dans les fichiers journaux du contrôleur : stderr et syslog. En outre, plusieurs tentatives de tâches sont exécutées simultanément, mais un fichier journal ne peut afficher les résultats que de manière linéaire.

Commencez par vérifier les journaux d'actions d'amorçage pour détecter les erreurs ou les modifications de configuration inattendues lors du lancement du cluster. À partir de là, consultez les journaux d'étapes pour identifier les tâches Hadoop lancées dans le cadre d'une étape comportant des erreurs. Examinez les journaux des tâches Hadoop pour identifier les tentatives de tâches qui ont échoué. Le journal des tentatives de tâche contiendra des détails sur la cause de l'échec d'une tentative de tâche.

Les sections suivantes décrivent comment utiliser les différents fichiers journaux pour identifier les erreurs 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 servent généralement à installer des logiciels supplémentaires sur le cluster ou à modifier les paramètres de configuration par rapport aux valeurs par défaut. La vérification des journaux peut fournir un aperçu des erreurs survenues lors de la configuration du cluster ainsi que des modifications des paramètres de configuration susceptibles d'attribuer les performances.

Vérification des journaux d'étape

Il existe quatre types de journaux d'étapes.

- Contrôleur : contient les fichiers générés par Amazon EMR (Amazon EMR) à la suite d'erreurs rencontrées lors de l'exécution de votre étape. Si votre étape échoue lors du 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 manquantes dans les fichiers de mappage.
- stderr : contient les messages d'erreur survenus lors du traitement de l'étape. Les erreurs de chargement des applications sont souvent décrites ici. Ce journal contient parfois une trace de pile.
- stdout : contient le statut généré par les exécutables de votre mappeur et de votre réducteur. Les erreurs de chargement des applications sont souvent décrites ici. Ce journal contient parfois des messages d'erreur d'application.
- syslog : contient des journaux provenant de logiciels autres qu'Amazon, tels qu'Apache et Hadoop. Les erreurs de diffusion sont souvent décrites ici.

Vérifiez stderr pour détecter les erreurs évidentes. Si stderr affiche une courte liste d'erreurs, l'étape s'est arrêtée rapidement et une erreur a été renvoyée. 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 syslog pour détecter les erreurs ou les défaillances. Suivez toutes les instructions concernant les tâches ayant échoué, en particulier si le message « Échec de la tâche » s'affiche.

Vérification des journaux de tentatives de tâche

Si l'analyse précédente des journaux d'étapes a révélé l'échec d'une ou de plusieurs tâches, examinez les journaux des tentatives de tâches correspondantes pour obtenir des informations plus détaillées sur les erreurs.

Étape 5 : Test du cluster étape par étape

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 plus d’informations, consultez Configuration d'un cluster pour qu'il continue ou se résilie après l'exécution de l'étape et Utilisation de la protection contre la résiliation.
2. Soumettez une étape au cluster. Pour plus d’informations, consultez Soumission de travail à un cluster.
3. À la fin du traitement de l'étape, recherchez les erreurs dans les fichiers journaux d'étape. Pour plus d’informations, consultez Étape 4 : Examiner les fichiers journaux. 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 Utilisation de la protection contre la résiliation.

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 Dépannage d'un cluster ayant échoué

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.

Les rubriques suivantes vous expliquent comment identifier d'autres causes possibles du ralentissement d'un cluster.

Rubriques

- Étape 1 : Rassembler des données sur le problème
- Étape 2 : Vérifier l'environnement
- Étape 3 : Examiner les fichiers journaux
- Étape 4 : Vérifier l'état du cluster et de l'instance
- Étape 5 : Vérifiez les groupes suspendus
- Étape 6 : Passer en revue les paramètres de configuration
- Étape 7 : Examiner les données d'entrée

Étape 1 : Rassembler des données sur le problème

La première étape du dépannage d'un cluster consiste à recueillir des informations sur le problème rencontré, ainsi que sur l'état actuel et la configuration du cluster. Ces informations seront utilisées dans les étapes suivantes pour confirmer ou exclure les causes possibles du problème.

Définition du problème

Définir clairement le problème est la première étape de résolution. Quelques questions à vous poser :

- Quel était l'effet attendu ? Que s'est-il passé à la place ?
- Quand ce problème s'est-il produit pour la première fois ? Combien de fois cela s'est-il produit depuis ?
- Est-ce que quelque chose a changé dans la façon dont je configure ou gère mon cluster ?

Détails du cluster

Les détails du cluster suivants sont utiles pour détecter les problèmes. Pour plus d'informations sur la manière de spécifier ces informations, consultez Afficher l'état et les détails d'un cluster.

- Identifiant du cluster. (Également appelé identifiant de flux de travail.)
- Région AWS et la zone de disponibilité dans laquelle le cluster a été lancé.
- État du cluster, y compris les détails du dernier changement d'état.
- Type et nombre d'instances EC2 spécifiées pour les nœuds principaux et de tâche.

Étape 2 : Vérifier l'environnement

Rubriques

- Recherche d'interruptions de service
- Recherche des limites d'utilisation
- Vérifiez la configuration du sous-réseau Amazon VPC
- Redémarrage du cluster

Recherche d'interruptions de service

Amazon EMR utilise plusieurs services Web Amazon en interne. Il exécute des serveurs virtuels sur Amazon EC2, stocke des données et des scripts sur Amazon S3 et transmet des métriques à. CloudWatch Les événements qui perturbent ces services sont rares, mais lorsqu'ils se produisent, ils peuvent entraîner des problèmes dans Amazon EMR.

Avant d'aller plus loin, consultez le Tableau de bord de l'état des services. Vérifiez la région dans laquelle vous avez lancé votre cluster pour voir s'il y a des interruptions dans l'un de ces services.

Recherche des limites d'utilisation

Si vous lancez un cluster de grande taille, si vous avez lancé plusieurs clusters simultanément ou si vous êtes un utilisateur partageant un cluster Compte AWS avec d'autres utilisateurs, le cluster a peut-être échoué car vous avez dépassé une limite de AWS service.

Amazon EC2 limite le nombre d'instances de serveurs virtuels exécutées dans une même AWS région à 20 instances réservées ou à la demande. Si vous lancez un cluster comportant plus de 20 nœuds, ou si vous lancez un cluster dont le nombre total d'instances EC2 actives dépasse 20, le cluster ne sera pas en mesure de lancer toutes les instances EC2 dont il a besoin et risque d'échouer. Compte AWS Dans ce cas, Amazon EMR renvoie une erreur `EC2 QUOTA EXCEEDED`. Vous pouvez demander à AWS augmenter le nombre d'instances EC2 que vous pouvez exécuter sur votre compte en soumettant une demande d'augmentation de la limite d'instance Amazon EC2.

Le délai entre la fermeture d'un cluster et le moment où il libère toutes ses ressources peut également vous faire dépasser vos limites d'utilisation. Selon sa configuration, il peut s'écouler entre 5 et 20 minutes avant que le cluster ne se termine complètement et ne libère les ressources qui lui ont été allouées. 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. 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 nouveau compartiment qui dépasse cette limite, la création du compartiment échouera et peut entraîner l'échec du cluster.

Vérifiez la configuration du sous-réseau Amazon VPC

Si votre cluster a été lancé dans un sous-réseau Amazon VPC, celui-ci doit être configuré comme décrit dans Configuration de la mise en réseau. Vérifiez également que le sous-réseau dans lequel vous lancez le cluster possède suffisamment d'adresses IP élastiques 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 trouver un code d'erreur ou une autre indication du problème rencontré par votre cluster. Pour plus d'informations sur les fichiers journaux disponibles, où les trouver et comment les consulter, consultez Afficher les fichiers journaux .

Il faudra peut-être effectuer un certain travail d'enquête pour déterminer ce qui s'est passé. Hadoop exécute le travail des tâches lors de tentatives de tâches sur différents nœuds du cluster. Amazon EMR peut lancer des tentatives de tâches spéculatives, mettant fin aux autres tentatives de tâches qui n'aboutissent pas. Cela génère une activité importante qui est enregistrée au fur et à mesure dans les fichiers journaux du contrôleur : stderr et syslog. En outre, plusieurs tentatives de tâches sont exécutées simultanément, mais un fichier journal ne peut afficher les résultats que de manière linéaire.

Commencez par vérifier les journaux d'actions d'amorçage pour détecter les erreurs ou les modifications de configuration inattendues lors du lancement du cluster. À partir de là, consultez les journaux d'étapes pour identifier les tâches Hadoop lancées dans le cadre d'une étape comportant des erreurs. Examinez les journaux des tâches Hadoop pour identifier les tentatives de tâches qui ont échoué. Le journal des tentatives de tâche contiendra des détails sur la cause de l'échec d'une tentative de tâche.

Les sections suivantes décrivent comment utiliser les différents fichiers journaux pour identifier les erreurs 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 servent généralement à installer des logiciels supplémentaires sur le cluster ou à modifier les paramètres de configuration par rapport aux valeurs par défaut. La vérification des journaux peut fournir un aperçu des erreurs survenues lors de la configuration du cluster ainsi que des modifications des paramètres de configuration susceptibles d'attribuer les performances.

Vérification des journaux d'étape

Il existe quatre types de journaux d'étapes.

- Contrôleur : contient les fichiers générés par Amazon EMR (Amazon EMR) à la suite d'erreurs rencontrées lors de l'exécution de votre étape. Si votre étape échoue lors du 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 manquantes dans les fichiers de mappage.

- stderr : contient les messages d'erreur survenus lors du traitement de l'étape. Les erreurs de chargement des applications sont souvent décrites ici. Ce journal contient parfois une trace de pile.
- stdout : contient le statut généré par les exécutables de votre mappeur et de votre réducteur. Les erreurs de chargement des applications sont souvent décrites ici. Ce journal contient parfois des messages d'erreur d'application.
- syslog : contient des journaux provenant de logiciels autres qu'Amazon, tels qu'Apache et Hadoop. Les erreurs de diffusion sont souvent décrites ici.

Vérifiez stderr pour détecter les erreurs évidentes. Si stderr affiche une courte liste d'erreurs, l'étape s'est arrêtée rapidement et une erreur a été renvoyée. 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 syslog pour détecter les erreurs ou les défaillances. Suivez toutes les instructions concernant les tâches ayant échoué, en particulier si le message « Échec de la tâche » s'affiche.

Vérification des journaux de tentatives de tâche

Si l'analyse précédente des journaux d'étapes a révélé l'échec d'une ou de plusieurs tâches, examinez les journaux des tentatives de tâches correspondantes pour obtenir des informations plus détaillées sur les erreurs.

Vérification des journaux démons Hadoop

Dans de rares cas, Hadoop lui-même peut échouer. Pour voir si c'est le cas, vous devez consulter les journaux Hadoop. Ils sont situés au niveau de `/var/log/hadoop/` sur chaque nœud.

Vous pouvez utiliser les JobTracker journaux pour associer une tentative de tâche infructueuse au nœud sur lequel elle a été exécutée. Une fois que vous connaissez le nœud rattaché à la tentative de tâche, vous pouvez vérifier l'état de l'instance EC2 qui héberge ce nœud pour voir s'il existe des problèmes tels que le manque de processeur ou de mémoire.

Étape 4 : Vérifier l'état du cluster et de l'instance

Un cluster Amazon EMR est composé de nœuds qui s'exécutent sur des instances Amazon EC2. 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 principal : gère le cluster. En cas de problème de performances, l'ensemble du cluster est attribué.
- nœuds principaux : traitent les tâches map-reduce et gèrent 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.
- nœuds de tâches : traitent les tâches map-reduce. 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.

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 :

Vérifiez 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. L'examen CloudWatch des indicateurs vous donne une vue d'ensemble de ce qui se passe dans votre cluster et peut vous donner un aperçu de la cause du ralentissement du traitement. Outre l'analyse CloudWatch d'un problème de performance existant, vous pouvez définir des alarmes qui CloudWatch déclenchent une alerte en cas de problème de performance futur. Pour plus d'informations, consultez Surveillance des métriques Amazon EMR avec CloudWatch.

Vérifier l'état de la tâche et l'état HDFS

Utilisez l'onglet Application user interfaces (Interfaces utilisateur d'application) sur la page des détails du cluster pour afficher les détails de l'application YARN. 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.

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.

- JobTracker — fournit des informations sur l'avancement de la tâche traitée 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 du HDFS et sur 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 de la tâche traitée 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 Afficher les instances de cluster dans Amazon EC2.

Étape 5 : Vérifiez les groupes suspendus

Un groupe d'instances est considéré « interrompu » quand il rencontre trop d'erreurs lorsqu'il tente de lancer des nœuds. Par exemple, si de nouveaux nœuds échouent à plusieurs reprises lors de l'exécution d'actions d'amorçage, au bout d'un certain temps, le groupe d'instances passera à l'état `SUSPENDED` plutôt que de tenter continuellement de mettre en service 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 l'état d'un groupe d'instances est `SUSPENDED` et si l'état du cluster est `WAITING`, vous pouvez ajouter une étape de cluster pour réinitialiser le nombre souhaité de nœuds principaux et de tâches.

L'ajout de l'étape déclenche la reprise du traitement du cluster et repasse l'état du groupe d'instance à `RUNNING`.

Pour plus d'informations sur la façon de réinitialiser un cluster dont l'état est interrompu, consultez État Interrompu.

Étape 6 : Passer en revue 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 à l'aide d'Amazon EMR, des paramètres spécifiques à Amazon EMR s'ajoutent aux 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 Création d'actions d'amorçage pour installer des logiciels supplémentaires. 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 des journaux sont stockées dans un fichier nommé `job_`*`job-id`*`_conf.xml` sous le répertoire `/mnt/var/log/hadoop/history/` du nœud principal, où *`job-id`* est remplacé par l'identifiant de la tâche. Si vous avez activé l'archivage des journaux, ces données sont copiées vers Amazon S3 dans le dossier `logs/`*`date`*`/`*`jobflow-id`*`/jobs`, où *`date`* est la date d'exécution de la tâche 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 leur impact sur le comportement de Hadoop, accédez à http://hadoop.apache.org/docs/.

⚠ Warning

1. Paramétrer `dfs.replication` sur la valeur 1 avec les clusters de moins de quatre nœuds peut entraîner une perte de données HDFS en cas de panne d'un seul nœud. Nous vous recommandons d'utiliser un cluster comportant au moins quatre nœuds principaux pour les charges de travail de production.
2. Amazon EMR n'autorisera pas les clusters à mettre à l'échelle les nœuds principaux situés en dessous de `dfs.replication`. Par exemple, si `dfs.replication = 2`, le nombre minimum de nœuds principaux est 2.
3. Lorsque vous utilisez la mise à l'échelle gérée, autoscaling, ou que vous choisissez de redimensionner manuellement votre cluster, nous vous recommandons de définir `dfs.replication` sur une valeur supérieure ou égale à 2.

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. Les options Java sont TaskTracker utilisées lors du lancement d'une machine virtuelle Java dans laquelle une tâche doit être exécutée. « -Xmx » est un paramètre courant pour définir la taille maximale de la mémoire.

Paramètre de configuration	Description
mapred.map.child.java.opts	Les options Java sont TaskTracker utilisées lors du lancement d'une machine virtuelle Java pour l'exécution d'une tâche cartographique. « -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	Les options Java sont TaskTracker utilisées lors du lancement d'une machine virtuelle Java pour 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 vous n'attendez pas assez longtemps, vous risquez de provoquer des erreurs « trop d'échecs de recherche » dans les 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.

Paramètre de configuration	Description
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.

Résoudre les problèmes liés à un cluster de Lake Formation

Cette section vous guide à travers le processus de dépannage des problèmes courants lors de l'utilisation d'Amazon EMR avec AWS Lake Formation.

L'accès au lac de données n'est pas autorisé

Vous devez explicitement opter pour le filtrage des données sur les clusters Amazon EMR avant de pouvoir analyser et traiter les données de votre lac de données. En cas d'échec de l'accès aux données, un message `Access is not allowed` générique apparaît dans la sortie des entrées de votre bloc-notes.

Pour activer et autoriser le filtrage des données sur Amazon EMR, consultez la section Autoriser le filtrage des données sur Amazon EMR du Guide du développeur AWS Lake Formation pour obtenir des instructions.

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 le tableau demandé

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 au tableau, vous devez accorder l'accès à l'utilisateur en mettant à jour les autorisations rattachées à ce tableau dans Lake Formation.

Interrogation de données entre comptes partagées avec Lake Formation

Lorsque vous utilisez Amazon EMR pour accéder aux données partagées avec vous depuis un autre compte, certaines bibliothèques Spark tentent d'appeler l'opération d'API `Glue:GetUserDefinedFunctions`. Les versions 1 et 2 des autorisations AWS RAM gérées ne prenant pas en charge cette action, le message d'erreur suivant s'affiche :

`"ERROR: User: arn:aws:sts::012345678901:assumed-role/my-spark-role/i-06ab8c2b59299508a is not authorized to perform: glue:GetUserDefinedFunctions on resource: arn:exampleCatalogResource because no resource-based policy allows the glue:GetUserDefinedFunctions action"`

Pour résoudre cette erreur, l'administrateur du lac de données qui a créé le partage de ressources doit mettre à jour les autorisations AWS RAM gérées associées au partage de ressources. La version 3 des autorisations gérées AWS RAM permet aux nœuds principaux d'effectuer l'action `glue:GetUserDefinedFunctions`.

Si vous créez un nouveau partage de ressources, Lake Formation applique la dernière version de l'autorisation AWS RAM gérée par défaut, et aucune action n'est requise de votre part. Pour activer l'accès aux données entre comptes pour les partages de ressources existants, vous devez mettre à jour les autorisations AWS RAM gérées vers la version 3.

Vous pouvez consulter les AWS RAM autorisations attribuées aux ressources partagées avec vous dans AWS RAM. Les autorisations suivantes sont incluses dans la version 3 :

```
Databases
  AWSRAMPermissionGlueDatabaseReadWriteForCatalog
  AWSRAMPermissionGlueDatabaseReadWrite

Tables
  AWSRAMPermissionGlueTableReadWriteForCatalog
  AWSRAMPermissionGlueTableReadWriteForDatabase

AllTables
  AWSRAMPermissionGlueAllTablesReadWriteForCatalog
  AWSRAMPermissionGlueAllTablesReadWriteForDatabase
```

Pour mettre à jour la version des autorisations AWS RAM gérées des partages de ressources existants

Vous (administrateur du lac de données) pouvez soit mettre à jour les autorisations AWS RAM gérées vers une version plus récente en suivant les instructions du guide de AWS RAM l'utilisateur, soit révoquer toutes les autorisations existantes pour le type de ressource et les réaccorder. Si vous révoquez les autorisations, le partage AWS RAM de AWS RAM ressources associé au type de ressource est supprimé. Lorsque vous réaccordez des autorisations, AWS RAM de nouveaux partages de ressources sont créés en y joignant la dernière version des autorisations AWS RAM gérées.

Insertion, création et modification de tableaux

L'insertion, la création ou la modification de tableaux dans des bases de données protégées par des politiques 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 avec. AWS Lake Formation

Écriture d'applications pour lancer et gérer des clusters

Rubriques

- Exemple de code source Java nd-to-end Amazon EMR
- Concepts communs pour les appels d'API
- Utiliser les SDK pour appeler les API Amazon EMR
- Gérer les Service Quotas Amazon EMR

Vous pouvez accéder aux fonctionnalités fournies par l'API Amazon EMR en appelant les fonctions wrapper dans l'un des SDK. AWS Les AWS SDK fournissent des fonctions spécifiques au langage qui intègrent l'API du service Web et simplifient la connexion au service Web, en gérant la plupart 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 Utiliser les SDK pour appeler les API Amazon EMR.

> **Important**
>
> Le taux de demandes maximal pour Amazon EMR est d'une demande toutes les dix secondes.

Exemple de code source Java nd-to-end Amazon EMR

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 décrit les end-to-end étapes nécessaires pour installer AWS Toolkit for Eclipse et exécuter un exemple de code source Java entièrement 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 Utiliser les SDK pour appeler les API Amazon EMR.

Cet exemple de code source Java montre comment effectuer les tâches suivantes à l'aide de l'API Amazon EMR :

- Récupérez les AWS informations d'identification et envoyez-les à Amazon EMR pour effectuer des appels d'API
- Configurer une nouvelle étape personnalisée et une nouvelle étape prédéfinie
- Ajouter de nouvelles étapes à un cluster Amazon EMR existant
- Récupérer les ID des étapes de cluster à partir d'un cluster en cours d'exécution

> **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 Téléchargements Eclipse.

Ensuite, installez le plug-in de développement de base de données pour Eclipse.

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 la boîte à outils pour Eclipse afin de rendre disponibles les modèles de projet de code source préconfigurés utiles.

Installer la boîte à outils pour 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 objets, choisissez AWS Toolkit for Eclipse puis Terminer.
5. Redémarrez Eclipse lorsque vous y êtes invité.

Créez ensuite un nouveau projet AWS Java et exécutez l'exemple de code source Java.

Pour créer un nouveau projet AWS Java

1. Ouvrez l'IDE Eclipse.
2. Choisissez File (Fichier), New (Nouveau) et Other (Autre).
3. Dans la boîte de dialogue Sélectionner un assistant, choisissez Projet Java AWS et Suivant.
4. Dans la boîte de dialogue Nouveau projet AWS Java, dans le **`Project name:`** champ, entrez le nom de votre nouveau projet, par exemple**`EMR-sample-code`**.
5. Choisissez Configurer les AWS comptes..., entrez vos clés d'accès publiques et privées, puis cliquez sur Terminer. Pour plus d'informations sur vos clés d'accès, consultez Comment puis-je obtenir les informations d'identification de sécurité ? dans le manuel Référence générale d'Amazon Web Services.

   > **Note**
   >
   > Vous ne devez pas 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 l'exemple de code suivant, remplacez l'exemple d'ID de cluster (JobFlowId) par un ID de cluster valide dans votre compte *j-xxxxxxxxxxxx*, qui se trouve soit dans le, AWS Management Console soit à l'aide de la AWS CLI commande suivante :
>
> ```
> aws emr list-clusters --active | grep "Id"
> ```
>
> En outre, remplacez l'exemple de chemin d'accès Amazon S3, *s3://path/to/my/jarfolder*, par le chemin d'accès valide de votre fichier JAR. Enfin, remplacez l'exemple de nom de classe, *com.my.Main1*, par le nom correct de la classe figurant dans votre fichier 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]
```

Concepts communs pour les appels d'API

Rubriques

- Points de terminaison pour Amazon EMR
- Spécification de paramètres de cluster dans Amazon EMR
- Zones de disponibilité dans Amazon EMR
- Comment utiliser des fichiers et des bibliothèques supplémentaires dans les clusters Amazon EMR

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.

Points de terminaison pour 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. Le point de terminaison indique la AWS région dans laquelle les clusters sont créés, décrits ou interrompus. Il a le format `elasticmapreduce.regionname.amazonaws.com`. Si vous spécifiez le point de terminaison 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 comptes plus anciens, la région par défaut est us-east-1.

Pour de plus amples informations sur les points de terminaison disponibles pour Amazon EMR, consultez Régions et points de terminaison dans le Référence générale d'Amazon Web Services.

Spécification de paramètres de cluster dans 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 résolvez les problèmes liés au

flux de travail et ajoutez des étapes alors que l'exécution du cluster est suspendue, définissez cette valeur sur `True`. Cela réduit le temps et les coûts requis pour charger les résultats vers Amazon Simple Storage Service (Amazon S3), uniquement pour répéter le processus après avoir modifié une étape pour redémarrer le cluster.

Si tel `KeepJobAlive` est le cas`true`, après avoir réussi à faire en sorte que le cluster termine son travail, vous devez envoyer une `TerminateJobFlows` demande, sinon le cluster continue de fonctionner et de générer des AWS frais.

Pour plus d'informations sur les paramètres propres à`RunJobFlow`, consultez RunJobFlow. Pour de plus amples informations sur les paramètres génériques dans la demande, consultez Paramètres de demande communs.

Zones de disponibilité dans 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 la liste des points de terminaison et des régions Amazon EMR disponibles, consultez 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 plus d'informations, consultez le guide de l'utilisateur Amazon EC2.

Comment utiliser des fichiers et des bibliothèques supplémentaires dans les clusters Amazon EMR

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.

Utiliser les SDK pour appeler les API Amazon EMR

Rubriques

- Utilisation du AWS SDK for Java pour créer un cluster Amazon EMR

Les AWS SDK fournissent des fonctions qui encapsulent l'API et prennent en charge de nombreux détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et la gestion des erreurs. Les 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 wrapper d'un SDK peut considérablement simplifier le processus d'écriture d'une AWS application.

Pour plus d'informations sur le téléchargement et l'utilisation des AWS SDK, consultez la section SDK sous Outils pour Amazon Web Services.

Utilisation du AWS SDK for Java pour créer un cluster Amazon EMR

AWS SDK for Java Il fournit trois packages dotés de la 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 la référence API AWS SDK for Java.

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.

```
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) {
  AWSCredentialsProvider profile = null;
  try {
   credentials_profile = new ProfileCredentialsProvider("default"); // 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(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 un rôle de flux de travail correspondant respectivement à EMR_ DefaultRole et DefaultRole EMR_EC2_. Vous pouvez le faire en appelant cette AWS CLI commande 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) seront 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 suivante pour les créer :

```
aws emr create-default-roles
```

Gérer les Service Quotas Amazon EMR

Rubriques

- Que sont les Service Quotas Amazon EMR
- Comment gérer les Service Quotas Amazon EMR
- Quand configurer des événements EMR dans CloudWatch

Les rubriques de cette section décrivent les quotas de service EMR (anciennement appelés limites de service), comment les gérer dans le AWS Management Console et dans quels cas il est avantageux d'utiliser des CloudWatch événements plutôt que des quotas de service pour surveiller les clusters et déclencher des actions.

Que sont les Service Quotas Amazon EMR

Votre AWS compte dispose de quotas de service par défaut, également appelés limites, pour chaque AWS service. Le service EMR comporte deux types de limites :

- Limites de ressources : vous pouvez utiliser l'EMR pour créer des ressources EC2. Toutefois, ces ressources EC2 sont soumises à des Service Quotas. Les limites de ressources dans cette catégorie sont les suivantes :
  - Nombre maximal de clusters actifs qui peuvent s'exécuter en même temps.
  - Le nombre maximum d'instances actives par groupe d'instances.
- Limites relatives aux API : lorsque vous utilisez des API EMR, les deux types de limites sont les suivants :

- Limite de rafale : il s'agit du nombre maximum d'appels d'API que vous pouvez effectuer simultanément. Par exemple, le nombre maximum de demandes d' AddInstanceFleet API que vous pouvez effectuer par seconde est fixé à 5 appels/seconde par défaut. Cela implique que la limite de rafale de AddInstanceFleet l'API est de 5 appels/seconde, ou que, à tout moment, vous pouvez effectuer au maximum 5 appels d' AddInstanceFleet API. Cependant, une fois que vous avez utilisé la limite de rafale, vos appels suivants sont limités par la limite de débit.
- Limite de débit : il s'agit de la capacité de débordement de la capacité de rafale de l'API. Par exemple, le taux de réapprovisionnement des AddInstanceFleet appels est défini par défaut à 0,5 appels/seconde. Cela signifie qu'une fois que vous avez atteint la limite de rafales, vous devez attendre au moins 2 secondes (0,5 appel/seconde X 2 secondes = 1 appel) pour effectuer l'appel d'API. Si vous passez un appel avant cela, vous êtes limité par le service Web EMR. À tout moment, vous ne pouvez passer qu'un nombre d'appels égal à la capacité de débordement sans être limité. Chaque seconde supplémentaire que vous attendez, votre capacité de débordement augmente de 0,5 appel jusqu'à ce qu'elle atteigne la limite maximale de 5, qui est la limite de débordement.

Comment gérer les Service Quotas Amazon EMR

Service Quotas est une AWS fonctionnalité que vous pouvez utiliser pour consulter et gérer vos quotas, ou limites, de service Amazon EMR depuis un emplacement central à l'aide de l' AWS Management Console API ou de la CLI. Pour plus d'informations sur les Service Quotas et la demande d'une augmentation, consultez AWS Service Quotas dans le Référence générale d'Amazon Web Services.

Pour certaines API, il peut être préférable de configurer un CloudWatch événement plutôt que d'augmenter les quotas de service. Vous pouvez également gagner du temps en configurant CloudWatch des alarmes et en déclenchant des demandes d'augmentation de manière proactive, avant d'atteindre le quota de service. Pour en savoir plus, consultez Quand configurer des événements EMR dans CloudWatch.

Quand configurer des événements EMR dans CloudWatch

Pour certaines API de sondage, telles que DescribeCluster DescribeStep, et ListClusters, la configuration d'un CouldWatch événement peut réduire le temps de réponse aux modifications et libérer vos quotas de service. Par exemple, si une fonction Lambda est configurée pour s'exécuter lorsque l'état d'un cluster change, par exemple lorsqu'une étape est terminée ou qu'un cluster se termine, vous pouvez utiliser ce déclencheur pour démarrer l'action suivante dans votre flux de

travail au lieu d'attendre le prochain sondage. Dans le cas contraire, si vous disposez d'instances Amazon EC2 dédiées ou de fonctions Lambda qui interrogent constamment l'API EMR pour détecter les modifications, vous gaspillez non seulement des ressources de calcul, mais vous risquez également d'atteindre votre Service Quota.

Voici quelques cas dans lesquels vous pourriez bénéficier du passage à une architecture axée sur les événements.

Cas 1 : Interrogation de l'EMR à l'aide d'appels d'DescribeCluster API pour terminer les étapes

Example Interroger l'EMR à l'aide d'appels d' DescribeCluster API pour terminer les étapes

Un modèle courant consiste à soumettre une étape à un cluster en cours d'exécution et à interroger Amazon EMR pour connaître le statut de l'étape, généralement à l'aide des API DescribeCluster or DescribeStep . Cette tâche peut également être accomplie dans un délai minimal en se connectant à l'événement Amazon EMR Step Status Change.

Cet événement inclut les informations suivantes dans sa charge utile.

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD
 (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```

Dans la carte détaillée, une fonction Lambda peut analyser « state », « StepID » ou « clusterID » pour trouver des informations pertinentes.

Cas 2 : Interrogation d'EMR sur les clusters disponibles pour exécuter des flux de travail

Example Interrogation d'EMR sur les clusters disponibles pour exécuter des flux de travail

Les clients qui exécutent plusieurs clusters ont pour habitude d'exécuter des flux de travail sur des clusters dès qu'ils sont disponibles. Si de nombreux clusters sont en cours d'exécution et qu'un flux de travail doit être exécuté sur un cluster en attente, une méthode peut consister à interroger l'EMR à l'aide d'appels EMR DescribeCluster ou d' ListClusters API pour connaître les clusters disponibles. Une autre façon de réduire le délai nécessaire pour savoir quand un cluster est prêt pour une étape serait de traiter l'événement de changement d'état du cluster Amazon EMR.

Cet événement inclut les informations suivantes dans sa charge utile.

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "WAITING",
    "message": "Amazon EMR cluster j-123456789ABCD ..."
  }
}
```

Pour cet événement, une fonction Lambda peut être configurée pour envoyer immédiatement un flux de travail en attente à un cluster dès que son statut passe à WAITING.

Cas 3 : Interrogation d'EMR pour mettre fin au cluster

Example Interrogation d'EMR pour mettre fin au cluster

Les clients qui gèrent de nombreux clusters EMR ont souvent tendance à interroger Amazon EMR pour savoir s'il s'agit de clusters interrompus afin que les tâches ne lui soient plus envoyées. Vous pouvez implémenter ce modèle avec les appels d' ListClusters API DescribeCluster et ou en utilisant l'événement Amazon EMR Cluster State Change dans.

Lorsque le cluster prend fin, l'événement émis ressemble à l'exemple suivant.

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has
 terminated at 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

La section « détail » de la charge utile inclut le clusterID et l'état sur lesquels il est possible d'agir.

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le Glossaire AWS dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.