aws

Guide du développeur

Amazon MemoryDB

Amazon MemoryDB: Guide du développeur

Table of Contents

Qu'est-ce que MemoryDB 1
Caractéristiques de MemoryDB 1
Composants principaux de MemoryDB 2
Clusters 3
Nœuds 4
Partitions 5
Groupes de paramètres 5
Groupes de sous-réseaux 5
Listes de contrôle d'accès (ACL) 6
Users 6
Services connexes 6
Choix des régions et zones de disponibilité 7
Localisation de vos nœuds 8
Régions et terminaux pris en charge 9
Accès à MemoryDB 12
Sécurité de MemoryDB 13
Commencer à utiliser MemoryDB 14
Configuration 14
Créez votre AWS compte 15
Octroi d’un accès par programmation 17
Configurez vos autorisations (nouveaux utilisateurs de MemoryDB uniquement) 18
Téléchargement et configuration de la AWS CLI 19
Étape 1 : créer un cluster 21
Création d'un cluster MemoryDB 21
Configuration de l'authentification 33
Étape 2 : Autoriser l'accès au cluster 34
Étape 3 : Connexion au cluster 36
Trouvez le point de terminaison de votre cluster 36
Se connecter à un cluster MemoryDB (Linux) 36
Étape 4 : Supprimer un cluster 38
Comment procéder ensuite ? 40
Gestion des nœuds 42
Nœuds et partitions MemoryDB 42
Types de nœuds pris en charge 44

Nœuds réservés ... 46
Vue d'ensemble des nœuds réservés ... 46
Remplacement de nœuds ... 58
Gestion des clusters ... 60
Mise à niveau des données ... 61
Bonnes pratiques ... 62
Limites ... 62
Mise à niveau de tarification des données ... 63
Surveillance ... 63
Utilisation de la hiérarchisation des données ... 63
Restauration des données d'un instantané vers des clusters avec la hiérarchisation des données activée ... 65
Préparation d'un cluster ... 67
Déterminer les exigences ... 67
Création d’un cluster ... 70
Affichage des détails d'un cluster ... 71
Modification d’un cluster ... 76
Ajouter/supprimer des nœuds d'un cluster ... 79
Accès à votre cluster ... 81
Accordez l'accès à votre cluster ... 81
Accès à MemoryDB depuis l'extérieur AWS ... 83
Recherche de points de terminaison de connexion ... 89
Partitions ... 92
Trouver le nom d'un shard ... 93
Gestion de votre implémentation de MemoryDB ... 97
Versions du moteur ... 97
Redis OSS 7.0 (amélioré) ... 97
Redis OSS 7.0 (amélioré) ... 98
Redis OSS 6.2 (amélioré) ... 99
Mise à niveau des versions de moteur ... 100
Mise en route avec JSON ... 102
Présentation du type de données JSON Redis OSS ... 103
Commandes prises en charge ... 115
Marquer vos ressources MemoryDB ... 157
Surveillance des coûts avec des balises ... 163
Gestion des balises à l'aide du AWS CLI ... 164

Gestion des balises à l'aide de l'API MemoryDB 168
Gestion de la maintenance 170
Bonnes pratiques 172
Commandes Redis OSS restreintes 173
Résilience 174
Bonnes pratiques : Pub/Sub et multiplexage d'E/S améliorées 176
Bonnes pratiques : redimensionnement des clusters en ligne 176
Comprendre la réplication MemoryDB 177
Cohérence 178
Réplication dans un cluster 178
Réduction des temps d'arrêt avec Multi-AZ 180
Modification du nombre de réplicas 188
Instantané et restauration 198
Constraints 199
Coûts 199
Planification de snapshots automatiques 201
Création d'instantanés manuels 202
Création d'un instantané final 205
Décrire les instantanés 207
Copie d'un instantané 210
Exportation d'un instantané 213
Restaurer à partir d'un instantané 223
Ensemencer un cluster avec un instantané 229
Marquage des instantanés 235
Suppression d'un instantané 236
Mise à l'échelle 237
Dimensionnement des clusters MemoryDB 239
Configuration des paramètres de moteur à l'aide de groupes de paramètres 262
Gestion des paramètres 263
Niveaux de groupe de paramètres 264
Création d'un groupe de paramètres 265
Liste des groupes de paramètres par nom 269
Affichage des valeurs d'un groupe de paramètres 274
Modification d'un groupe de paramètres 275
Suppression d'un groupe de paramètres 278
Paramètres spécifiques à Redis OSS 280

Tutoriel : Configuration d'une fonction Lambda pour accéder à MemoryDB dans un Amazon VPC 298
Étape 1 : créer un cluster 298
Étape 2 : créer une fonction Lambda 301
Étape 3 : Tester la fonction Lambda 305
Étape 4 : Nettoyage (facultatif) 306
Recherche vectorielle 308
Aperçu de la recherche vectorielle 308
Index et espaces clés 309
Types de champs d'index 310
Algorithmes d'index vectoriel 311
Expression de requête de recherche vectorielle 312
INFO commande 315
Sécurité de la recherche vectorielle 317
Cas d’utilisation 318
Génération augmentée de récupération () RAG 318
Cache sémantique durable 319
Détection des fraudes 320
Autres cas d’utilisation 321
Caractéristiques et limites de la recherche vectorielle 321
Disponibilité de la recherche vectorielle 321
Restrictions paramétriques 321
Limites d'échelle 322
Restrictions opérationnelles 322
Importation/exportation de snapshots et migration en direct 323
Consommation de mémoire 323
Mémoire insuffisante pendant le remblayage 327
Transactions 327
En utilisant le AWS Management Console 327
En utilisant le AWS Command Line Interface 328
Commandes de recherche vectorielle 328
PIEDS. CREATE 329
PIEDS. SEARCH 333
PIEDS. AGGREGATE 336
PIEDS. DROPINDEX 337
PIEDS. INFO 338

PIEDS. _ LIST ... 340
PIEDS. ALIASADD ... 341
PIEDS. ALIASDEL ... 341
PIEDS. ALIASUPDATE ... 341
PIEDS. _ ALIASLIST ... 342
PIEDS. PROFILE ... 342
PIEDS. EXPLAIN ... 342
PIEDS. EXPLAINCLI ... 343
Sécurité ... 344
Protection des données ... 345
Sécurité des données dans MemoryDB ... 346
Chiffrement au repos ... 347
Chiffrement en transit (TLS) ... 350
Authentification des utilisateurs à l'aide des ACL ... 351
Authentification avec IAM ... 366
Gestion des identités et des accès ... 374
Public ciblé ... 374
Authentification par des identités ... 375
Gestion des accès à l’aide de politiques ... 379
Comment fonctionne MemoryDB avec IAM ... 382
Exemples de politiques basées sur l’identité ... 392
Résolution des problèmes ... 395
Contrôle d’accès ... 397
Présentation de la gestion des accès ... 399
Journalisation et surveillance ... 428
Surveillance avec CloudWatch ... 429
Surveillance des événements ... 449
Journalisation des appels d'API MemoryDB avec AWS CloudTrail ... 463
Validation de conformité ... 470
Sécurité de l’infrastructure ... 471
Confidentialité du trafic inter-réseau ... 471
MemoryDB et Amazon VPC et Amazon VPC ... 472
Sous-réseaux et groupes de sous-réseaux ... 485
API MemoryDB et points de terminaison VPC d'interface ()AWS PrivateLink ... 499
Mises à jour de service ... 503
Gestion des mises à jour du service ... 503

Référence 507
Utilisation de l'API MemoryDB 508
Utilisation de l'API Query 508
Bibliothèques disponibles 511
Applications de dépannage 512
Quotas 514
Historique de la documentation 515
..... dxix

Qu'est-ce que MemoryDB

MemoryDB est un service de base de données en mémoire durable qui fournit des performances ultrarapides. Il est spécialement conçu pour les applications modernes dotées d'architectures de microservices.

MemoryDB est compatible avec Redis OSS, un magasin de données open source populaire, qui vous permet de créer rapidement des applications en utilisant les mêmes structures de données, API et commandes Redis OSS flexibles et conviviales que celles qu'ils utilisent déjà aujourd'hui. Avec MemoryDB, toutes vos données sont stockées en mémoire, ce qui vous permet d'atteindre une microseconde en lecture, une latence d'écriture d'un chiffre en millisecondes et un débit élevé. MemoryDB stocke également les données de manière durable dans plusieurs zones de disponibilité (AZ) à l'aide d'un journal transactionnel multi-AZ pour permettre un basculement rapide, la restauration de bases de données et le redémarrage des nœuds.

Offrant à la fois des performances en mémoire et une durabilité multi-AZ, MemoryDB peut être utilisée comme base de données principale haute performance pour vos applications de microservices, éliminant ainsi le besoin de gérer séparément une base de données cache et une base de données durable.

Rubriques

- Caractéristiques de MemoryDB
- Composants principaux de MemoryDB
- Services connexes
- Choix des régions et zones de disponibilité
- Accès à MemoryDB
- Sécurité de MemoryDB

Caractéristiques de MemoryDB

MemoryDB est un service de base de données en mémoire durable qui fournit des performances ultrarapides. Les fonctionnalités de MemoryDB incluent :

- Forte cohérence pour les nœuds principaux et cohérence finale garantie pour les nœuds répliques. Pour plus d’informations, consultez Cohérence.

- Latences de lecture en microsecondes et en écriture d'un chiffre en millisecondes avec un maximum de 160 millions de TPS par cluster.
- Structures de données et API Redis OSS flexibles et conviviales. Créez facilement de nouvelles applications ou migrez des applications Redis OSS existantes sans pratiquement aucune modification.
- Durabilité des données grâce à un journal transactionnel multi-AZ permettant une restauration et un redémarrage rapides de la base de données.
- Disponibilité multi-AZ avec basculement automatique, détection des défaillances des nœuds et restauration en cas de défaillance.
- Effectuez facilement une mise à l'échelle horizontale en ajoutant et en supprimant des nœuds ou verticalement en passant à des types de nœuds plus ou moins grands. Vous pouvez augmenter le débit d'écriture en ajoutant des partitions et le débit de lecture en ajoutant des répliques.
- ead-after-write Cohérence R pour les nœuds principaux et cohérence finale garantie pour les nœuds répliques.
- MemoryDB prend en charge le chiffrement en transit, le chiffrement au repos et l'authentification des utilisateurs via. Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL)
- Instantanés automatiques dans Amazon S3 avec conservation jusqu'à 35 jours.
- Support pour un maximum de 500 nœuds et plus de 100 To de stockage par cluster (avec une réplique par partition).
- Chiffrement en transit avec TLS et chiffrement au repos avec clés. AWS KMS
- Authentification et autorisation des utilisateurs avec Redis OSSAuthentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).
- Support pour les types d'instances AWS Graviton2.
- Intégration à d'autres AWS services tels qu' CloudWatchAmazon VPC et Amazon SNS pour la surveillance CloudTrail, la sécurité et les notifications.
- Correctifs logiciels et mises à niveau entièrement gérés.
- AWS Intégration de Identity and Access Management (IAM) et contrôle d'accès basé sur des balises pour les API de gestion.

Composants principaux de MemoryDB

Vous trouverez ci-dessous un aperçu des principaux composants d'un déploiement de MemoryDB.

Rubriques

- Clusters
- Nœuds
- Partitions
- Groupes de paramètres
- Groupes de sous-réseaux
- Liste de contrôle d'accès (ACL)
- Users

Clusters

Un cluster est un ensemble d'un ou plusieurs nœuds desservant un seul ensemble de données. Un ensemble de données MemoryDB est partitionné en partitions, et chaque partition possède un nœud principal et jusqu'à 5 nœuds de réplication facultatifs. Un nœud principal traite les demandes de lecture et d'écriture, tandis qu'une réplique ne traite que les demandes de lecture. Un nœud principal peut basculer vers un nœud de réplique, promouvant ainsi cette réplique au rang de nouveau nœud principal pour cette partition. MemoryDB exécute Redis OSS comme moteur de base de données, et lorsque vous créez un cluster, vous spécifiez la version de Redis OSS pour votre cluster. Vous pouvez créer et modifier un cluster à l' AWS CLI aide de l'API MemoryDB ou du. AWS Management Console

Chaque cluster MemoryDB exécute une version du moteur Redis OSS. Chaque version du moteur Redis OSS possède ses propres fonctionnalités prises en charge. En outre, chaque version du moteur Redis OSS possède un ensemble de paramètres dans un groupe de paramètres qui contrôlent le comportement des clusters qu'elle gère.

La capacité de calcul et de mémoire d'un cluster est déterminée par son type de nœud. Vous pouvez sélectionner le type de nœud qui correspond le mieux à vos besoins. Si vos besoins évoluent au fil du temps, vous pouvez modifier les types de nœuds. Pour plus d’informations, veuillez consulter Types de nœuds pris en charge.

> **Note**
>
> Pour obtenir des informations sur les tarifs des types de nœuds MemoryDB, consultez la section Tarification de MemoryDB.

Vous exécutez un cluster sur un cloud privé virtuel (VPC) à l'aide du service Amazon Virtual Private Cloud (Amazon VPC). Lorsque vous utilisez un VPC, vous disposez d'un contrôle total sur l'environnement de réseau virtuel. Vous pouvez choisir votre propre plage d'adresses IP, créer des sous-réseaux et configurer le routage et les listes de contrôle d'accès. MemoryDB gère les instantanés, les correctifs logiciels, la détection automatique des défaillances et la restauration. Il n'y a pas de frais supplémentaires pour exécuter votre cluster dans un VPC. Pour plus d'informations sur l'utilisation d'Amazon VPC avec MemoryDB, consultez. MemoryDB et Amazon VPC et Amazon VPC

De nombreuses opérations MemoryDB ciblent les clusters :

- Création d’un cluster
- Modification d’un cluster
- Prendre des instantanés d'un cluster
- Suppression d’un cluster
- Affichage des éléments d'un cluster
- Ajout ou suppression des balises de répartition des coûts vers et depuis un cluster

Pour en savoir plus, consultez les rubriques connexes suivantes :

- Gestion des clusters et Gestion des nœuds

  Informations sur les clusters, les nœuds et les opérations connexes.
- Résilience dans MemoryDB

  Informations sur l'amélioration de la tolérance aux pannes de vos clusters.

Nœuds

Un nœud est le plus petit élément constitutif d'un déploiement de MemoryDB et s'exécute à l'aide d'une instance Amazon EC2. Chaque nœud exécute la version Redis OSS choisie lors de la création de votre cluster. Un nœud appartient à une partition appartenant à un cluster.

Chaque nœud exécute une instance du moteur dans la version choisie lors de la création de votre cluster. Si nécessaire, vous pouvez redimensionner les nœuds d'un cluster vers le haut ou vers le bas pour obtenir un type différent. Pour plus d’informations, consultez Mise à l'échelle .

Chaque nœud d'un cluster est du même type. Plusieurs types de nœuds sont pris en charge, chacun ayant une quantité de mémoire variable. Pour obtenir la liste des types de nœuds pris en charge, consultez Types de nœuds pris en charge.

Pour plus d'informations sur les nœuds, consultez Gestion des nœuds.

Partitions

Une partition est un regroupement de un à 6 nœuds, l'un servant de nœud d'écriture principal et les 5 autres servant de répliques de lecture. Un cluster MemoryDB possède toujours au moins une partition.

Les clusters MemoryDB peuvent contenir jusqu'à 500 partitions, vos données étant partitionnées entre les partitions. Par exemple, vous pouvez choisir de configurer un cluster de 500 nœuds compris entre 83 (un principal et 5 réplicas par partition) et 500 partitions (un principal et aucun réplicas). Assurez-vous qu'il y ait suffisamment d'adresses IP disponibles pour faire face à l'augmentation. Les pièges courants incluent les sous-réseaux du groupe de sous-réseaux avec une plage CIDR trop petite ou les sous-réseaux partagés et fortement utilisés par d'autres clusters.

Une partition avec plusieurs nœuds implémente la réplication avec un nœud principal en lecture/écriture et de 1 à 5 nœuds de réplica. Pour plus d'informations, consultez Comprendre la réplication MemoryDB.

Pour plus d'informations sur les partitions, consultez Utilisation de partitions.

Groupes de paramètres

Les groupes de paramètres constituent un moyen simple de gérer les paramètres d'exécution de Redis OSS sur votre cluster. Les paramètres sont utilisés pour contrôler l'utilisation de la mémoire, la taille des éléments, etc. Un groupe de paramètres MemoryDB est un ensemble nommé de paramètres spécifiques au moteur que vous pouvez appliquer à un cluster, et tous les nœuds de ce cluster sont configurés exactement de la même manière.

Pour des informations plus détaillées sur les groupes de paramètres MemoryDB, consultez. Configuration des paramètres de moteur à l'aide de groupes de paramètres

Groupes de sous-réseaux

Un groupe de sous-réseaux est un ensemble de sous-réseaux (généralement privés) que vous pouvez utiliser pour vos clusters fonctionnant dans un environnement Amazon Virtual Private Cloud (VPC).

Lorsque vous créez un cluster dans un Amazon VPC, vous pouvez spécifier un groupe de sous-réseaux ou utiliser celui fourni par défaut. MemoryDB utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds.

Pour des informations plus détaillées sur les groupes de sous-réseaux MemoryDB, consultez. Sous-réseaux et groupes de sous-réseaux

Liste de contrôle d'accès (ACL)

Une liste de contrôle d'accès est un ensemble d'un ou de plusieurs utilisateurs. Les chaînes d'accès suivent les règles de l'ACL Redis OSS pour autoriser l'accès des utilisateurs aux commandes et aux données de Redis OSS.

Pour des informations plus détaillées sur les listes de contrôle d'accès MemoryDB, consultez. Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL)

Users

Un utilisateur possède un nom d'utilisateur et un mot de passe et est utilisé pour accéder aux données et émettre des commandes sur votre cluster MemoryDB. Un utilisateur est membre d'une liste de contrôle d'accès (ACL), que vous pouvez utiliser pour déterminer les autorisations de cet utilisateur sur les clusters MemoryDB. Pour de plus amples informations, veuillez consulter Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).

Services connexes

ElastiCache (Redis OSS)

Lorsque vous décidez d'utiliser MemoryDB ou ElastiCache (Redis OSS), considérez les comparaisons suivantes :

- MemoryDB est une base de données en mémoire durable pour les charges de travail nécessitant une base de données principale ultrarapide. Vous devriez envisager d'utiliser MemoryDB si votre charge de travail nécessite une base de données durable offrant des performances ultra-rapides (latence en microsecondes en lecture et de l'ordre de la milliseconde en écriture). MemoryDB peut également convenir à votre cas d'utilisation si vous souhaitez créer une application à l'aide de structures de données et d'API Redis OSS avec une base de données principale durable. Enfin, vous devriez envisager d'utiliser MemoryDB pour simplifier l'architecture de votre application et

réduire les coûts en remplaçant l'utilisation d'une base de données par un cache pour garantir la durabilité et les performances.

- ElastiCache (Redis OSS) est un service couramment utilisé pour mettre en cache des données provenant d'autres bases de données et magasins de données à l'aide de Redis OSS. Vous devriez envisager ElastiCache (Redis OSS) de mettre en cache les charges de travail lorsque vous souhaitez accélérer l'accès aux données avec votre base de données principale ou votre magasin de données existant (performances de lecture et d'écriture en microsecondes). Vous devez également envisager ElastiCache (Redis OSS) les cas d'utilisation dans lesquels vous souhaitez utiliser les structures de données et les API Redis OSS pour accéder aux données stockées dans une base de données ou un magasin de données principal.

Choix des régions et zones de disponibilité

AWS Les ressources de cloud computing sont hébergées dans des centres de données hautement disponibles. Pour offrir une évolutivité et une fiabilité supplémentaires, ces installations de centre de données sont situées dans différents emplacements physiques. Ces emplacements sont classés par régions et zones de disponibilité.

AWS Les régions sont vastes et largement dispersées dans des zones géographiques distinctes. Les zones de disponibilité sont des emplacements distincts au sein d'une AWS région conçus pour être isolés des défaillances dans d'autres zones de disponibilité. Ils fournissent une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même AWS région.

> **Important**
>
> Chaque région est totalement indépendante. Toute activité MemoryDB que vous lancez (par exemple, la création de clusters) s'exécute uniquement dans votre région par défaut actuelle.

Pour créer ou utiliser un cluster dans une région spécifique, utilisez le point de terminaison du service régional correspondant. Pour les points de terminaison de service, consultez Régions et terminaux pris en charge.

Localisation de vos nœuds

Tout cluster comportant au moins une réplique doit être réparti sur plusieurs zones de disponibilité. La seule façon de tout localiser au sein d'une seule zone de zone est d'utiliser un cluster composé de partitions à nœud unique.

En localisant les nœuds dans différentes zones de zone, MemoryDB élimine le risque qu'une panne, telle qu'une panne de courant, dans une zone de zone entraîne une perte de disponibilité.

- Création d'un cluster MemoryDB
- Modification d'un cluster MemoryDB

Régions et terminaux pris en charge

MemoryDB est disponible dans plusieurs AWS régions. Cela signifie que vous pouvez lancer des clusters MemoryDB dans des emplacements qui répondent à vos besoins. Par exemple, vous pouvez lancer votre produit dans la AWS région la plus proche de vos clients ou dans une AWS région spécifique pour répondre à certaines exigences légales. De plus, à mesure que MemoryDB étend la disponibilité à une nouvelle AWS région, MemoryDB prend en charge les deux `MAJOR.MINOR` versions les plus récentes à ce moment-là pour la nouvelle région. Pour plus d'informations sur les versions de MemoryDB, consultez. Versions du moteur Redis OSS

Par défaut, les AWS SDK AWS CLI, l'API MemoryDB et la console MemoryDB font référence à la région US-Est (Virginie du Nord). À mesure que MemoryDB étend la disponibilité à de nouvelles régions, de nouveaux points de terminaison pour ces régions peuvent également être utilisés dans vos requêtes HTTP, les AWS SDK et la console AWS CLI.

Chaque région est conçue pour être complètement isolée des autres régions . Chaque région dispose de plusieurs zones de disponibilité (AZ). En lançant vos nœuds dans différentes zones de disponibilité, vous obtenez la plus grande tolérance aux pannes possible. Pour plus d'informations sur les régions et les zones de disponibilité, reportez-vous Choix des régions et zones de disponibilité au début de cette rubrique.

Régions où MemoryDB est pris en charge

Nom de région/Région	Point de terminaison	Protocole	
Région US East (Ohio) `us-east-2`	`memory-db.us-east-2.amazonaws.com`	HTTPS	
Région USA Est (Virginie du Nord) `us-east-1`	`memory-db.us-east-1.amazonaws.com`	HTTPS	
Région US West (N. California) `us-west-1`	`memory-db.us-west-1.amazonaws.com`	HTTPS	

Nom de région/Région	Point de terminaison	Protocole	
Région USA Ouest (Oregon) `us-west-2`	`memory-db.us-west-2.amazonaws.com`	HTTPS	
Région Canada (Centre) `ca-central-1`	`memory-db.ca-central-1.amazonaws.com`	HTTPS	
Région Asie-Pacifique (Hong Kong) `ap-east-1`	`memory-db.ap-eastl-1.amazonaws.com`	HTTPS	
Région Asie-Pacifique (Mumbai) `ap-south-1`	`memory-db.ap-south-1.amazonaws.com`	HTTPS	
Région Asie-Pacifique (Tokyo) `ap-northeast-1`	`memory-db.ap-northeast-1.amazonaws.com`	HTTPS	
Région Asia Pacific (Seoul) `ap-northeast-2`	`memory-db.ap-northeast-2.amazonaws.com`	HTTPS	
Région Asie-Pacifique (Singapour) `ap-southeast-1`	`memory-db.ap-southeast-1.amazonaws.com`	HTTPS	

Nom de région/Région	Point de terminaison	Protocole	
Région Asie-Pacifique (Sydney) `ap-southeast-2`	`memory-db.ap-southeast-2.amazonaws.com`	HTTPS	
Région Europe (Francfort) `eu-central-1`	`memory-db.eu-central-1.amazonaws.com`	HTTPS	
Région Europe (Irlande) `eu-west-1`	`memory-db.eu-west-1.amazonaws.com`	HTTPS	
Région Europe (Londres) `eu-west-2`	`memory-db.eu-west-2.amazonaws.com`	HTTPS	
Région EU (Paris) `eu-west-3`	`memory-db.eu-west-3.amazonaws.com`	HTTPS	
Région Europe (Stockholm) `eu-north-1`	`memory-db.eu-north-1.amazonaws.com`	HTTPS	
Europe (Milan) Region `eu-south-1`	`memory-db.eu-south-1.amazonaws.com`	HTTPS	

Nom de région/Région	Point de terminaison	Protocole	
Région Amérique du Sud (São Paulo) `sa-east-1`	`memory-db.sa-east-1.amazonaws.com`	HTTPS	
Région Chine (Beijing) `cn-north-1`	`memory-db.cn-north-1.amazonaws.com.cn`	HTTPS	
Région Chine (Ningxia) `cn-northwest-1`	`memory-db.cn-northwest-1.amazonaws.com.cn`	HTTPS	

Pour un tableau des AWS produits et services par région, voir Produits et services par région.

Pour un tableau des zones de disponibilité prises en charge au sein des régions, voirSous-réseaux et groupes de sous-réseaux.

Accès à MemoryDB

Chaque point de terminaison du cluster MemoryDB contient une adresse et un port. Ce point de terminaison du cluster prend en charge le protocole Redis OSS Cluster pour permettre aux clients de découvrir les rôles, adresses IP et emplacements spécifiques pour chaque nœud du cluster. Lorsqu'un nœud principal tombe en panne et qu'une réplique est promue à sa place, vous pouvez vous connecter au point de terminaison du cluster pour découvrir le nouveau nœud principal à l'aide du protocole Redis OSS Cluster.

Vous devez vous connecter au point de terminaison du cluster pour découvrir les points de terminaison du nœud à l'aide de la cluster slots commande cluster nodes or. Après avoir découvert le nœud approprié pour une clé, vous pouvez vous connecter directement au nœud pour les demandes de lecture/écriture. Un client Redis OSS peut utiliser le point de terminaison du cluster pour se connecter automatiquement au nœud approprié.

Pour dépanner des nœuds spécifiques d'un cluster, vous pouvez également utiliser des points de terminaison spécifiques à un nœud, mais ceux-ci ne sont pas nécessaires pour une utilisation normale.

Pour trouver le point de terminaison d'un cluster, consultez les rubriques suivantes :

- Trouver le point de terminaison d'un cluster MemoryDB (CLI)AWS
- Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Pour la connexion à des nœuds ou à des clusters, consultezConnexion aux nœuds MemoryDB à l'aide de redis-cli.

Sécurité de MemoryDB

La sécurité de MemoryDB est gérée à trois niveaux :

- Pour contrôler qui peut effectuer des actions de gestion sur les clusters et les nœuds MemoryDB, vous utilisez AWS Identity and Access Management (IAM). Lorsque vous vous connectez à AWS l'aide d'informations d'identification IAM, votre AWS compte doit disposer de politiques IAM qui accordent les autorisations requises pour effectuer des opérations. Pour plus d’informations, consultez Gestion des identités et des accès dans MemoryDB.
- Pour contrôler les niveaux d'accès aux clusters, vous créez des utilisateurs dotés d'autorisations spécifiées et vous les assignez aux listes de contrôle d'accès (ACL). L'ACL, à son tour, est ensuite associée à un ou plusieurs clusters. Pour plus d’informations, consultez Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).
- Les clusters MemoryDB doivent être créés dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC. Pour contrôler quels appareils et instances Amazon EC2 peuvent ouvrir des connexions au point de terminaison et au port du nœud pour les clusters MemoryDB dans un VPC, vous utilisez un groupe de sécurité VPC. Avec ces points de terminaison et les connexions de port, vous pouvez utiliser TLS/SSL (Transport Layer Security/Secure Sockets Layer). En outre, les règles de pare-feu de votre entreprise peuvent contrôler si les appareils fonctionnant dans votre entreprise peuvent ouvrir des connexions à un cluster MemoryDB. Pour plus d'informations sur les VPC, consultez MemoryDB et Amazon VPC et Amazon VPC.

Pour plus d'informations sur la configuration de la sécurité, consultez Sécurité dans MemoryDB.

Commencer à utiliser MemoryDB

Cet exercice explique les étapes à suivre pour créer, autoriser l'accès, s'y connecter et enfin supprimer un cluster MemoryDB à l'aide de la console de gestion MemoryDB.

> **Note**
>
> Dans le cadre de cet exercice, nous vous recommandons d'utiliser l'option Easy create lors de la création d'un cluster et de revenir aux deux autres options une fois que vous aurez exploré plus en détail les fonctionnalités de MemoryDB.

Rubriques

- Configuration
- Étape 1 : créer un cluster
- Étape 2 : Autoriser l'accès au cluster
- Étape 3 : Connexion au cluster
- Étape 4 : Supprimer un cluster
- Comment procéder ensuite ?

Configuration

Vous trouverez ci-dessous des rubriques qui décrivent les actions ponctuelles que vous devez effectuer pour commencer à utiliser MemoryDB.

Rubriques

- Créez votre AWS compte
- Octroi d'un accès par programmation
- Configurez vos autorisations (nouveaux utilisateurs de MemoryDB uniquement)
- Téléchargement et configuration de la AWS CLI

Créez votre AWS compte

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 AWS services 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.

Octroi d’un accès par programmation

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

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

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

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

Voir aussi:

- Qu'est-ce qu'IAM ? dans le guide de l'utilisateur IAM.
- AWS Informations d'identification de sécurité dans la référence AWS générale.

Configurez vos autorisations (nouveaux utilisateurs de MemoryDB uniquement)

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .

- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) du Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM du Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

MemoryDB crée et utilise des rôles liés à des services pour fournir des ressources et accéder à d'autres AWS ressources et services en votre nom. Pour que MemoryDB crée un rôle lié à un service pour vous, utilisez la AWS politique -managed nommée. `AmazonMemoryDBFullAccess` Ce rôle est préconfiguré avec l'autorisation que le service requiert pour créer un rôle lié au service en votre nom.

Vous pouvez décider de ne pas utiliser la politique par défaut et d'utiliser une politique gérée personnalisée. Dans ce cas, assurez-vous que vous êtes autorisé à appeler `iam:createServiceLinkedRole` ou que vous avez créé le rôle lié au service MemoryDB.

Pour plus d'informations, consultez les ressources suivantes :

- Création d'une politique (IAM)
- AWS-politiques gérées (prédéfinies) pour MemoryDB
- Utilisation de rôles liés à un service pour MemoryDB

Téléchargement et configuration de la AWS CLI

AWS CLI Il est disponible à l'adresse http://aws.amazon.com/cli. Elle s'exécute sous Windows, macOS et Linux. Après avoir téléchargé le AWS CLI, procédez comme suit pour l'installer et le configurer :

1. Consultez le Guide de l'utilisateur de l'interface de ligne de commande AWS.
2. Suivez les instructions d'installation de la AWS CLI et de configuration de la AWS CLI.

Étape 1 : créer un cluster

Avant de créer un cluster pour une utilisation en production, vous devez évidemment réfléchir à la façon dont vous allez configurer le cluster pour répondre aux besoins métier. Ces questions sont abordées dans la section Préparation d'un cluster. Dans le cadre de cet exercice de mise en route, vous pouvez accepter les valeurs de configuration par défaut lorsqu'elles s'appliquent.

Le cluster que vous allez créer sera opérationnel, et non pas exécuté dans un environnement de test (sandbox). Vous devrez payer les frais d'utilisation standard de MemoryDB pour l'instance jusqu'à ce que vous la supprimiez. Le total frais seront minimes (généralement moins d'un dollar) si vous terminez l'exercice décrit ici en une seule fois et que vous supprimez votre cluster quand vous avez terminé. Pour plus d'informations sur les taux d'utilisation de MemoryDB, consultez MemoryDB.

Votre cluster est lancé dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC.

Création d'un cluster MemoryDB

Les exemples suivants montrent comment créer un cluster à l'aide de l'API AWS Management Console, AWS CLI et MemoryDB.

Création d'un cluster (console)

Pour créer un cluster à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Choisissez Clusters dans le volet de navigation de gauche, puis choisissez Create.

Easy create

1. Renseignez la section Configuration. Cela permet de configurer le type de nœud et la configuration par défaut de votre cluster. Sélectionnez la taille de mémoire et les performances réseau appropriées dont vous avez besoin parmi les options suivantes :
   - Production
   - Développement/Test
   - Démonstration
2. Complétez la section Informations sur le cluster.

a. Dans Nom, entrez un nom pour votre cluster.

   Les contraintes d'attribution de noms de cluster sont les suivantes :

   - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
   - Doit commencer par une lettre.
   - Ils ne peuvent pas comporter deux traits d'union consécutifs.
   - Ils ne peuvent pas se terminer par un trait d'union.

b. Dans la zone Description, entrez une description du cluster.

3. Complétez la section Groupes de sous-réseaux :

   - Pour les groupes de sous-réseaux, créez un nouveau groupe de sous-réseaux ou choisissez-en un existant dans la liste disponible que vous souhaitez appliquer à ce cluster. Si vous en créez un nouveau :

     - Entrez un nom
     - Entrez une description
     - Si vous avez activé Multi-AZ, le groupe de sous-réseaux doit contenir au moins deux sous-réseaux résidant dans des zones de disponibilité différentes. Pour plus d'informations, consultez Sous-réseaux et groupes de sous-réseaux.
     - Si vous créez un nouveau groupe de sous-réseaux et que vous n'avez pas de VPC existant, il vous sera demandé de créer un VPC. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

4. Pour la recherche vectorielle, vous pouvez activer la fonction de recherche vectorielle pour stocker les intégrations vectorielles et effectuer des recherches vectorielles. Notez que cela corrigera les valeurs relatives à la compatibilité des versions de Redis OSS, aux groupes de paramètres et aux partitions. Pour plus d'informations, consultez Recherche vectorielle.

5. Afficher les paramètres par défaut :

   Lorsque vous utilisez Easy create, les paramètres de cluster restants sont définis par défaut. Notez que certains de ces paramètres peuvent être modifiés après la création, comme indiqué par Modifiable après la création.

6. Pour les tags, vous pouvez éventuellement appliquer des tags pour rechercher et filtrer vos clusters ou suivre vos AWS coûts.

7. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create pour lancer votre cluster ou Cancel pour annuler l'opération.

Dès que l'état de votre cluster est disponible, vous pouvez accorder un accès EC2, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 2 : Autoriser l'accès au cluster.

> **Important**
>
> Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Create new cluster

1. Complétez la section Informations sur le cluster.
   a. Dans Nom, entrez un nom pour votre cluster.

      Les contraintes d'attribution de noms de cluster sont les suivantes :

      - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
      - Doit commencer par une lettre.
      - Ils ne peuvent pas comporter deux traits d'union consécutifs.
      - Ils ne peuvent pas se terminer par un trait d'union.

   b. Dans la zone Description, entrez une description du cluster.
2. Complétez la section Groupes de sous-réseaux :

   - Pour les groupes de sous-réseaux, créez un nouveau groupe de sous-réseaux ou choisissez-en un existant dans la liste disponible que vous souhaitez appliquer à ce cluster. Si vous en créez un nouveau :

     - Entrez un nom
     - Entrez une description

- Si vous avez activé Multi-AZ, le groupe de sous-réseaux doit contenir au moins deux sous-réseaux résidant dans des zones de disponibilité différentes. Pour plus d'informations, consultez Sous-réseaux et groupes de sous-réseaux.
- Si vous créez un nouveau groupe de sous-réseaux et que vous n'avez pas de VPC existant, il vous sera demandé de créer un VPC. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

3. Complétez la section Paramètres du cluster :

   a. Pour activer la fonctionnalité de recherche vectorielle, vous pouvez l'activer pour stocker des intégrations vectorielles et effectuer des recherches vectorielles. Notez que cela corrigera les valeurs relatives à la compatibilité des versions de Redis OSS, aux groupes de paramètres et aux partitions. Pour plus d'informations, consultez Recherche vectorielle.

   b. Pour la compatibilité des versions de Redis OSS, acceptez la valeur par défaut`6.2`.

   c. Pour Port, acceptez le port Redis OSS par défaut 6379 ou, si vous avez une raison d'utiliser un autre port, entrez le numéro de port.

   d. Pour le groupe de paramètres, si vous avez activé la recherche vectorielle, utilisez`default.memorydb-redis7.search.preview`. Dans le cas contraire, acceptez le groupe de `default.memorydb-redis7` paramètres.

      Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres spécifiques à Redis OSS.

   e. Pour Type de nœud, choisissez une valeur pour le type de nœud (ainsi que la taille de mémoire associée) que vous souhaitez.

      Si vous choisissez un type de nœud de la famille r6gd, vous activerez automatiquement la hiérarchisation des données, qui divise le stockage de données entre la mémoire et le SSD. Pour plus d'informations, consultez Mise à niveau des données.

   f. Dans Nombre de partitions, choisissez le nombre de partitions que vous souhaitez pour ce cluster. Pour une meilleure disponibilité de vos clusters, nous vous recommandons d'ajouter au moins 2 partitions.

      Vous pouvez modifier le nombre de partitions de votre cluster de manière dynamique. Pour plus d'informations, consultez Dimensionnement des clusters MemoryDB.

g. Pour Réplicas par partition, choisissez le nombre de nœuds de réplica en lecture souhaité dans chaque partition.

   Les restrictions suivantes existent :

   - Si Multi-AZ est activé, assurez-vous d'avoir au moins un réplica par partition.
   - Le nombre de réplicas est le même pour chaque partition lors de la création du cluster à l'aide de la console.

h. Choisissez Next (Suivant)

i. Complétez la section Paramètres avancés :

   i. Pour Groupes de sécurité, choisissez les groupes de sécurité que vous souhaitez utiliser pour ce cluster. Un groupe de sécurité agit comme un pare-feu pour contrôler l'accès réseau à votre cluster. Vous pouvez utiliser le groupe de sécurité par défaut pour votre VPC ou en créer un nouveau.

      Pour plus d'informations sur les groupes de sécurité, consultez Groupes de sécurité pour votre VPC dans le Guide de l'utilisateur Amazon VPC.

   ii. Pour le chiffrement de vos données, vous avez les options suivantes :

      - Encryption at rest (Chiffrement au repos) : active le chiffrement des données stockées sur le disque. Pour de plus amples informations, veuillez consulter Chiffrement au repos.

        > **Note**
        >
        > Vous avez la possibilité de fournir une clé de chiffrement autre que celle par défaut en choisissant la clé KMS AWS gérée par le client et en choisissant la clé.

      - Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Si vous ne sélectionnez aucun chiffrement, une liste de contrôle d'accès ouverte appelée « accès ouvert » sera créée avec un utilisateur par défaut. Pour plus d'informations, consultez Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).

iii. Pour Snapshot, spécifiez éventuellement une période de conservation des instantanés et une fenêtre de capture d'écran. Par défaut, l'option Activer les instantanés automatiques est présélectionnée.

iv. Pour la fenêtre de maintenance, spécifiez éventuellement une fenêtre de maintenance. La fenêtre de maintenance est la période, généralement d'une heure, pendant laquelle MemoryDB planifie la maintenance du système pour votre cluster chaque semaine. Vous pouvez autoriser MemoryDB à choisir le jour et l'heure de votre fenêtre de maintenance (aucune préférence), ou vous pouvez choisir vous-même le jour, l'heure et la durée (Spécifiez la fenêtre de maintenance). Si vous choisissez Specify maintenance window, choisissez dans les listes les valeurs de Start day, Start time et Duration (en heures) pour le créneau de maintenance. Toutes les heures sont en UTC.

Pour plus d'informations, consultez Gestion de la maintenance.

v. Pour Notifications, choisissez une rubrique Amazon Simple Notification Service (Amazon SNS) existante ou choisissez une entrée ARN manuelle et tapez l'Amazon Resource Name (ARN) de la rubrique. Amazon SNS permet d'émettre des notifications push vers des appareils connectés à Internet. La valeur par défaut consiste à désactiver les notifications. Pour plus d'informations, consultez https://aws.amazon.com/sns/.

vi. Pour les tags, vous pouvez éventuellement appliquer des tags pour rechercher et filtrer vos clusters ou suivre vos AWS coûts.

j. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create pour lancer votre cluster ou Cancel pour annuler l'opération.

Dès que l'état de votre cluster est disponible, vous pouvez accorder un accès EC2, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 2 : Autoriser l'accès au cluster.

> **Important**
>
> Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne

plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Restore from snapshots

Sous Source du cliché, choisissez le cliché source à partir duquel vous souhaitez migrer les données. Pour plus d'informations, consultez Instantané et restauration .

> **Note**
>
> Si vous souhaitez que la recherche vectorielle soit activée dans votre nouveau cluster, la recherche vectorielle doit également être activée sur l'instantané source.

Le cluster cible utilise par défaut les paramètres du cluster source. Vous pouvez éventuellement modifier les paramètres suivants sur le cluster cible :

1. Informations sur le cluster
   a. Dans Nom, entrez un nom pour votre cluster.

      Les contraintes d'attribution de noms de cluster sont les suivantes :

      - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
      - Doit commencer par une lettre.
      - Ils ne peuvent pas comporter deux traits d'union consécutifs.
      - Ils ne peuvent pas se terminer par un trait d'union.

   b. Dans la zone Description, entrez une description du cluster.
2. Groupes de sous-réseaux
   - Pour les groupes de sous-réseaux, créez un nouveau groupe de sous-réseaux ou choisissez-en un existant dans la liste disponible que vous souhaitez appliquer à ce cluster. Si vous en créez un nouveau :
     - Entrez un nom
     - Entrez une description

- Si vous avez activé Multi-AZ, le groupe de sous-réseaux doit contenir au moins deux sous-réseaux résidant dans des zones de disponibilité différentes. Pour plus d'informations, consultez Sous-réseaux et groupes de sous-réseaux.
- Si vous créez un nouveau groupe de sous-réseaux et que vous n'avez pas de VPC existant, il vous sera demandé de créer un VPC. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

3. Paramètres du cluster

   a. Pour activer la fonctionnalité de recherche vectorielle, vous pouvez l'activer pour stocker des intégrations vectorielles et effectuer des recherches vectorielles. Notez que cela corrigera les valeurs relatives à la compatibilité des versions de Redis OSS, aux groupes de paramètres et aux partitions. Pour plus d'informations, consultez Recherche vectorielle.

   b. Pour la compatibilité des versions de Redis OSS, acceptez la valeur par défaut`6.2`.

   c. Pour Port, acceptez le port Redis OSS par défaut 6379 ou, si vous avez une raison d'utiliser un autre port, entrez le numéro de port.

   d. Pour le groupe de paramètres, si vous avez activé la recherche vectorielle, utilisez`default.memorydb-redis7.search.preview`. Dans le cas contraire, acceptez le groupe de `default.memorydb-redis7` paramètres.

      Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres spécifiques à Redis OSS.

   e. Pour Type de nœud, choisissez une valeur pour le type de nœud (ainsi que la taille de mémoire associée) que vous souhaitez.

      Si vous choisissez un type de nœud de la famille r6gd, vous activerez automatiquement la hiérarchisation des données, qui divise le stockage de données entre la mémoire et le SSD. Pour plus d'informations, consultez Mise à niveau des données.

   f. Dans Nombre de partitions, choisissez le nombre de partitions que vous souhaitez pour ce cluster. Pour une meilleure disponibilité de vos clusters, nous vous recommandons d'ajouter au moins 2 partitions.

      Vous pouvez modifier le nombre de partitions de votre cluster de manière dynamique. Pour plus d'informations, consultez Dimensionnement des clusters MemoryDB.

g. Pour Réplicas par partition, choisissez le nombre de nœuds de réplica en lecture souhaité dans chaque partition.

   Les restrictions suivantes existent :

   - Si Multi-AZ est activé, assurez-vous d'avoir au moins un réplica par partition.
   - Le nombre de réplicas est le même pour chaque partition lors de la création du cluster à l'aide de la console.

h. Choisissez Next (Suivant)

i. Réglages avancés

   i. Pour Groupes de sécurité, choisissez les groupes de sécurité que vous souhaitez utiliser pour ce cluster. Un groupe de sécurité agit comme un pare-feu pour contrôler l'accès réseau à votre cluster. Vous pouvez utiliser le groupe de sécurité par défaut pour votre VPC ou en créer un nouveau.

      Pour plus d'informations sur les groupes de sécurité, consultez Groupes de sécurité pour votre VPC dans le Guide de l'utilisateur Amazon VPC.

   ii. Pour le chiffrement de vos données, vous avez les options suivantes :

      - Encryption at rest (Chiffrement au repos) : active le chiffrement des données stockées sur le disque. Pour de plus amples informations, veuillez consulter Chiffrement au repos.

        > **Note**
        >
        > Vous avez la possibilité de fournir une clé de chiffrement autre que celle par défaut en choisissant la clé KMS AWS gérée par le client et en choisissant la clé.

      - Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Si vous ne sélectionnez aucun chiffrement, une liste de contrôle d'accès ouverte appelée « accès ouvert » sera créée avec un utilisateur par défaut. Pour plus d'informations, consultez Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).

iii. Pour Snapshot, spécifiez éventuellement une période de conservation des instantanés et une fenêtre de capture d'écran. Par défaut, l'option Activer les instantanés automatiques est présélectionnée.

iv. Pour la fenêtre de maintenance, spécifiez éventuellement une fenêtre de maintenance. La fenêtre de maintenance est la période, généralement d'une heure, pendant laquelle MemoryDB planifie la maintenance du système pour votre cluster chaque semaine. Vous pouvez autoriser MemoryDB à choisir le jour et l'heure de votre fenêtre de maintenance (aucune préférence), ou vous pouvez choisir vous-même le jour, l'heure et la durée (Spécifiez la fenêtre de maintenance). Si vous choisissez Specify maintenance window, choisissez dans les listes les valeurs de Start day, Start time et Duration (en heures) pour le créneau de maintenance. Toutes les heures sont en UTC.

Pour plus d'informations, consultez Gestion de la maintenance.

v. Pour Notifications, choisissez une rubrique Amazon Simple Notification Service (Amazon SNS) existante ou choisissez une entrée ARN manuelle et tapez l'Amazon Resource Name (ARN) de la rubrique. Amazon SNS permet d'émettre des notifications push vers des appareils connectés à Internet. La valeur par défaut consiste à désactiver les notifications. Pour plus d'informations, consultez https://aws.amazon.com/sns/.

vi. Pour les tags, vous pouvez éventuellement appliquer des tags pour rechercher et filtrer vos clusters ou suivre vos AWS coûts.

j. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create pour lancer votre cluster ou Cancel pour annuler l'opération.

Dès que l'état de votre cluster est disponible, vous pouvez accorder un accès EC2, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 2 : Autoriser l'accès au cluster.

> **Important**
>
> Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne

plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Création d'un cluster (AWS CLI)

Pour créer un cluster à l'aide du AWS CLI, voir `create-cluster`. Voici un exemple :

Pour Linux, macOS ou Unix :

```
aws memorydb create-cluster \
    --cluster-name my-cluster \
    --node-type db.r6g.large \
    --acl-name my-acl \
    --subnet-group my-sg
```

Pour Windows :

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6g.large ^
   --acl-name my-acl ^
   --subnet-group my-sg
```

Vous devriez obtenir la réponse JSON suivante :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
```

```
        "ACLName": "my-acl",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Vous pouvez commencer à utiliser le cluster une fois que son statut passe à`available`.

> **Important**
>
> Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Création d'un cluster (API MemoryDB)

Pour créer un cluster à l'aide de l'API MemoryDB, utilisez l'CreateClusteraction.

> **Important**
>
> Dès que votre cluster est disponible, vous serez facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous l'utilisez pas. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Configuration de l'authentification

Pour plus d'informations sur la configuration de l'authentification pour votre cluster, reportez-vous Authentification avec IAM aux sections etAuthentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).

Étape 2 : Autoriser l'accès au cluster

Cette section part du principe que vous savez lancer des instances Amazon EC2 et établir des connexions à ces instances. Pour plus d'informations, consultez le Guide de démarrage Amazon EC2.

Les clusters MemoryDB sont conçus pour être accessibles depuis une instance Amazon EC2. Ils sont également accessibles par des applications conteneurisées ou sans serveur exécutées dans Amazon Elastic Container Service ou. AWS Lambda Le scénario le plus courant consiste à accéder à un cluster MemoryDB à partir d'une instance Amazon EC2 dans le même Amazon Virtual Private Cloud (Amazon VPC), ce qui sera le cas pour cet exercice.

Avant de vous connecter à un cluster à partir d'une instance EC2, vous devez autoriser l'instance EC2 à accéder au cluster.

Le cas d'utilisation le plus courant concerne une application déployée sur une instance EC2 qui doit se connecter à un cluster du même VPC. La solution la plus simple pour gérer l'accès entre les instances EC2 et les clusters du même VPC consiste à agir ainsi :

1. Créez un groupe de sécurité VPC pour votre cluster. Ce groupe de sécurité peut être utilisé pour restreindre l'accès aux clusters. Par exemple, vous pouvez créer une règle personnalisée pour ce groupe de sécurité, qui autorise l'accès TCP à l'aide du port que vous avez attribué au cluster lorsque vous l'avez créé et une adresse IP que vous utiliserez pour accéder au cluster.

   Le port par défaut pour les clusters MemoryDB est. 6379
2. Créez un groupe de sécurité VPC pour vos instances EC2 (serveurs web et d'application). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l'instance EC2 à partir d'Internet via la table de routage du VPC. Par exemple, vous pouvez définir des règles sur ce groupe de sécurité pour autoriser l'accès TCP à l'instance EC2 sur le port 22.
3. Créez des règles personnalisées dans le groupe de sécurité de votre cluster qui autorisent les connexions à partir du groupe de sécurité que vous avez créé pour vos instances EC2. N'importe quel membre du groupe de sécurité peut ainsi accéder aux clusters.

Pour créer une règle dans un groupe de sécurité VPC qui autorise les connexions à partir d'un autre groupe de sécurité

1. Connectez-vous à la console de AWS gestion et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc.

2. Dans le volet de navigation de gauche, sélectionnez Security Groups.
3. Sélectionnez ou créez un groupe de sécurité que vous utiliserez pour vos clusters. Sous Règles entrantes, sélectionnez Modifier les règles entrantes, puis Ajouter une règle. Ce groupe de sécurité autorisera l'accès aux membres d'un autre groupe de sécurité.
4. Dans Type, choisissez Règle TCP personnalisée.

    a. Pour Plage de ports, spécifiez le port utilisé lors de la création de votre cluster.

    Le port par défaut pour les clusters MemoryDB est. 6379

    b. Dans le champ Source, saisissez l'ID de votre groupe de sécurité. Dans la liste, sélectionnez le groupe de sécurité que vous utiliserez pour vos instances Amazon EC2.
5. Choisissez Enregistrer lorsque vous avez terminé.

Une fois que vous avez activé l'accès, vous êtes prêt à vous connecter au cluster, comme indiqué dans la section suivante.

Pour plus d'informations sur l'accès à votre cluster MemoryDB à partir d'un autre Amazon VPC, d'une autre AWS région ou même de votre réseau d'entreprise, consultez ce qui suit :

- Modèles d'accès pour accéder à un cluster MemoryDB dans un Amazon VPC
- Accès aux ressources de MemoryDB depuis l'extérieur AWS

Étape 3 : Connexion au cluster

Avant de continuer, terminez la section Étape 2 : Autoriser l'accès au cluster.

Cette section suppose que vous avez créé une instance Amazon EC2 et que vous pouvez vous y connecter. Pour obtenir des instructions sur la façon de procéder, consultez le Guide de démarrage Amazon EC2.

Une instance Amazon EC2 ne peut se connecter à un cluster que si vous l'y autorisez.

Trouvez le point de terminaison de votre cluster

Une fois que votre cluster a l'état available (disponible) et que vous avez autorisé l'accès à ce cluster, vous pouvez vous connecter à une instance Amazon EC2 et vous connecter au cluster. Pour cela, vous devez d'abord déterminer le point de terminaison.

Pour en savoir plus sur la manière de trouver vos points de terminaison, consultez les rubriques suivantes :

- Trouver le point de terminaison d'un cluster MemoryDB ()AWS Management Console
- Trouver le point de terminaison d'un cluster MemoryDB (CLI)AWS
- Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Se connecter à un cluster MemoryDB (Linux)

Maintenant que vous disposez du point de terminaison dont vous avez besoin, vous pouvez vous connecter à une instance EC2 et au cluster. Dans l'exemple suivant, vous utilisez l'utilitaire cli pour vous connecter à un cluster à l'aide d'Ubuntu 22. La dernière version de cli prend également en charge le protocole SSL/TLS pour connecter les clusters compatibles avec le chiffrement/ l'authentification.

Connexion aux nœuds MemoryDB à l'aide de redis-cli

Pour accéder aux données depuis les nœuds MemoryDB, vous utilisez des clients qui fonctionnent avec le protocole SSL (Secure Socket Layer). Vous pouvez également utiliser redis-cli avec TLS/SSL sur Amazon Linux et Amazon Linux 2.

Pour utiliser redis-cli pour vous connecter à un cluster MemoryDB sur Amazon Linux 2 ou Amazon Linux

1. Téléchargez et compilez l'utilitaire redis-cli. Cet utilitaire est inclus dans la distribution du logiciel Redis OSS.
2. À l'invite de commande de votre instance EC2, tapez les commandes appropriées pour la version de Linux que vous utilisez.

   Amazon Linux 2023

   Si vous utilisez Amazon Linux 2023, saisissez ce qui suit :

   ```
   sudo yum install redis6 -y
   ```

   Tapez ensuite la commande suivante, en remplaçant le point de terminaison et le port de votre cluster par ceux illustrés dans cet exemple.

   ```
   redis-cli -h Primary or Configuration Endpoint --tls -p 6379
   ```

   Pour plus d'informations sur la recherche du point de terminaison, veuillez consulter Rechercher vos points de terminaison de nœud.

   Amazon Linux 2

   Si vous utilisez Amazon Linux 2, entrez ceci :

   ```
   sudo yum -y install openssl-devel gcc
   wget http://download.redis.io/redis-stable.tar.gz
   tar xvzf redis-stable.tar.gz
   cd redis-stable
   make distclean
   make redis-cli BUILD_TLS=yes
   sudo install -m 755 src/redis-cli /usr/local/bin/
   ```

   Amazon Linux

   Si vous utilisez Amazon Linux, saisissez ce qui suit :

   ```
   sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
   wget http://download.redis.io/redis-stable.tar.gz
   ```

```
tar xvzf redis-stable.tar.gz
cd redis-stable
make redis-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/redis-cli /usr/local/bin/
```

Sur Amazon Linux, vous pouvez également avoir besoin de suivre les étapes suivantes :

```
sudo yum install clang
CC=clang make
sudo make install
```

3. Après avoir téléchargé et installé l'utilitaire redis-cli, il est recommandé d'exécuter la commande facultative. `make-test`
4. Pour vous connecter à un cluster avec le chiffrement et l'authentification activés, entrez cette commande :

```
redis-cli -h Primary or Configuration Endpoint --tls -a 'your-password' -p 6379
```

> **Note**
>
> Si vous installez redis6 sur Amazon Linux 2023, vous pouvez désormais utiliser `redis6-cli` la commande au lieu de : `redis-cli`
>
> ```
> redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
> ```

Étape 4 : Supprimer un cluster

Tant que l'état d'un cluster est available, ce cluster vous est facturé, que vous l'utilisiez activement ou pas. Pour ne plus être facturé, supprimez le cluster.

> **Warning**
>
> Lorsque vous supprimez un cluster MemoryDB, vos instantanés manuels sont conservés. Vous pouvez également créer un instantané final avant la suppression du cluster. Les instantanés automatiques ne sont pas conservés. Pour plus d'informations, consultez Instantané et restauration .

À l'aide du AWS Management Console

La procédure suivante supprime un cluster unique de votre déploiement. Pour supprimer plusieurs clusters, répétez la procédure pour chaque cluster à supprimer. Vous n'avez pas besoin d'attendre la fin de la suppression d'un cluster avant de démarrer la procédure pour en supprimer un autre.

Pour supprimer un cluster

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Pour choisir le cluster à supprimer, cliquez sur le bouton radio situé à côté du nom du cluster dans la liste des clusters. Dans ce cas, le nom du cluster que vous avez créé sur Étape 1 : créer un cluster.
3. Pour Actions, choisissez Supprimer.
4. Choisissez d'abord de créer un instantané du cluster avant de le supprimer, puis entrez `delete` dans la case de confirmation et cliquez sur Supprimer pour supprimer le cluster, ou choisissez Annuler pour conserver le cluster.

   Si vous choisissez Delete, le cluster passe à l'état Suppression en cours.

Dès que votre cluster n'est plus répertorié dans la liste des clusters, il n'est plus facturé.

À l'aide du AWS CLI

Le code suivant supprime le cluster `my-cluster`. Dans ce cas, remplacez `my-cluster` par le nom du cluster que vous avez créé sur Étape 1 : créer un cluster.

```
aws memorydb delete-cluster --cluster-name my-cluster
```

L'opération `delete-cluster` CLI ne supprime qu'un seul cluster. Pour supprimer plusieurs clusters, appelez `delete-cluster` chaque cluster que vous souhaitez supprimer. Il n'est pas nécessaire d'attendre la fin de la suppression d'un cluster pour en supprimer un autre.

Pour Linux, macOS ou Unix :

```
aws memorydb delete-cluster \
    --cluster-name my-cluster \
    --region us-east-1
```

Pour Windows :

```
aws memorydb delete-cluster ^
    --cluster-name my-cluster ^
    --region us-east-1
```

Pour plus d'informations, consultez `delete-cluster`.

Utilisation de l'API MemoryDB

Le code suivant supprime le cluster `my-cluster`. Dans ce cas, remplacez `my-cluster` par le nom du cluster que vous avez créé sur Étape 1 : créer un cluster.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteCluster
    &ClusterName=my-cluster
    &Region=us-east-1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210802T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210802T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

L'opération `DeleteCluster` d'API ne supprime qu'un seul cluster. Pour supprimer plusieurs clusters, appelez `DeleteCluster` chaque cluster que vous souhaitez supprimer. Il n'est pas nécessaire d'attendre la fin de la suppression d'un cluster pour en supprimer un autre.

Pour plus d'informations, consultez DeleteCluster.

Comment procéder ensuite ?

Maintenant que vous avez essayé l'exercice Getting Started, vous pouvez explorer les sections suivantes pour en savoir plus sur MemoryDB et les outils disponibles :

- Commencer avec AWS
- Outils pour Amazon Web Services

- Interface de ligne de commande AWS
- Référence de l'API MemoryDB.

Gestion des nœuds

Un nœud est le plus petit élément constitutif d'un déploiement de MemoryDB. Un nœud appartient à une partition appartenant à un cluster. Chaque nœud exécute la version du moteur choisie lors de la création ou de la dernière modification du cluster. Chaque nœud a son propre port et nom DNS (Domain Name Service). Plusieurs types de nœuds MemoryDB sont pris en charge, chacun étant associé à des quantités variables de mémoire et de puissance de calcul.

Rubriques

- Nœuds et partitions MemoryDB
- Types de nœuds pris en charge
- Nœuds réservés MemoryDB
- Remplacement de nœuds

Voici certaines opérations importantes impliquant des nœuds :

- Ajouter/supprimer des nœuds d'un cluster
- Mise à l'échelle
- Recherche de points de terminaison de connexion

Nœuds et partitions MemoryDB

Une partition est un arrangement hiérarchique de nœuds, chacun étant encapsulé dans un cluster. Les partitions prennent en charge la réplication. Au sein d'une partition, un nœud fonctionne comme le nœud primaire de lecture/écriture. Tous les autres nœuds contenus dans une partition fonctionnent comme des réplicas en lecture seule du nœud primaire. MemoryDB prend en charge plusieurs partitions au sein d'un cluster. Cette prise en charge permet de partitionner vos données dans un cluster MemoryDB.

MemoryDB prend en charge la réplication via des partitions. Le fonctionnement de l'API DescribeClustersrépertorie les partitions avec les nœuds membres, les noms des nœuds, les points de terminaison et également d'autres informations.

Une fois qu'un cluster MemoryDB est créé, il peut être modifié (redimensionné ou réduit). Pour plus d’informations, consultez Mise à l'échelle et Remplacement de nœuds.

Lorsque vous créez un nouveau cluster, vous pouvez l'alimenter avec des données de l'ancien cluster afin qu'il ne démarre pas vide. Cela peut être utile si vous devez modifier le type de nœud, la version du moteur ou effectuer une migration depuis Amazon ElastiCache (Redis OSS). Pour plus d’informations, consultez Création d'instantanés manuels et Restaurer à partir d'un instantané.

Types de nœuds pris en charge

MemoryDB prend en charge les types de nœuds suivants.

Mémoire optimisée

Type d'instance	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)	Multiplexage E/S amélioré (Redis OSS 7.0.4+)	Version minimale du moteur
db.r7g.large	0,937	12,5	Non	6.2
db.r7g.xlarge	1,876	12,5	Non	6.2
db.r7g.2xlarge	3,75	15	Oui	6.2
db.r7g.4xlarge	7,5	15	Oui	6.2
db.r7g.8xlarge	15	N/A	Oui	6.2
db.r7g.12xlarge	22,5	N/A	Oui	6.2
db.r7g.16xlarge	30	N/A	Oui	6.2
db.r6g.large	0.75	10,0	Non	6.2
db.r6g.xlarge	1,25	10,0	Non	6.2
db.r6g.2xlarge	2,5	10,0	Oui	6.2
db.r6g.4xlarge	5.0	10,0	Oui	6.2
db.r6g.8xlarge	12	N/A	Oui	6.2
db.r6g.12xlarge	20	N/A	Oui	6.2
db.r6g.16xlarge	25	N/A	Oui	6.2

Mémoire optimisée avec la hiérarchisation des données

Type d'instance	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)	Multiplexage E/S amélioré (Redis OSS 7.0.4+)	Version minimale du moteur
db.r6gd.xlarge	1,25	10	Non	6.2
db.r6g.2xlarge	2,5	10	Non	6.2
db.r6g.4xlarge	5.0	10	Non	6.2
db.r6g.8xlarge	12	N/A	Non	6.2

Nœuds à usage général

Type d'instance	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)	Multiplexage E/S amélioré (Redis OSS 7.0.4+)	Version minimale du moteur
db.t4g.small	0,128	5.0	Non	6.2
db.t4g.medium	0,256	5.0	Non	6.2

Pour connaître AWS la disponibilité par région, consultez la tarification de MemoryDB

Tous les types de nœuds sont créés dans un cloud privé virtuel (VPC).

Nœuds réservés MemoryDB

Les nœuds réservés vous offrent une réduction significative par rapport à la tarification des nœuds à la demande. Les nœuds réservés ne sont pas des nœuds physiques, mais plutôt une réduction de facturation appliquée à l'utilisation de nœuds à la demande dans votre compte. Les remises pour les nœuds réservés sont liées au type de nœud et à AWS la région.

Le processus général d'utilisation des nœuds réservés est le suivant :

- Consultez les informations sur les offres de nœuds réservés disponibles
- Achetez une offre de nœuds réservés à l'aide du AWS Management Console AWS Command Line Interface ou du SDK
- Consultez les informations relatives à vos nœuds réservés existants

Rubriques

- Vue d'ensemble des nœuds réservés

Vue d'ensemble des nœuds réservés

Lorsque vous achetez un nœud réservé MemoryDB, vous vous engagez à bénéficier d'un tarif réduit, sur un type de nœud spécifique, pendant toute la durée du nœud réservé. Pour utiliser un nœud réservé MemoryDB, vous devez créer un nouveau nœud comme vous le feriez pour un nœud à la demande. Le nouveau nœud que vous créez doit correspondre aux spécifications du nœud réservé. Si les spécifications du nouveau nœud correspondent à celles d'un nœud réservé existant pour votre compte, vous êtes facturé au tarif réduit proposé pour le nœud réservé. Dans le cas contraire, le nœud est facturé au tarif à la demande. Vous pouvez utiliser l'API AWS Management Console AWS CLI, la ou l'API MemoryDB pour répertorier et acheter les offres de nœuds réservés disponibles.

MemoryDB propose des nœuds réservés pour les nœuds R7g, R6g et R6gd (avec hiérarchisation des données) optimisés pour la mémoire. Pour plus d'informations sur les tarifs, consultez la section Tarification de MemoryDB.

Types d'offres

Les nœuds réservés sont disponibles en trois types (aucun initial, initial partiel et total initial) qui vous permettent d'optimiser les coûts de MemoryDB en fonction de votre utilisation prévue.

Pas de paiement initial : cette option permet d'accéder à un nœud réservé sans nécessiter de paiement initial. Votre nœud réservé No Upfront facture un tarif horaire réduit pour chaque heure pendant le terme, quelle que soit l'utilisation, et aucun paiement initial n'est requis.

Montant initial partiel : cette option nécessite le paiement initial d'une partie du nœud réservé. Les heures restantes pendant la période sont facturées à un taux réduit, quelle que soit l'utilisation.

Tout à l'avance — Le paiement intégral est effectué au début du terme, aucun autre coût n'étant encouru pendant le reste du terme, quel que soit le nombre d'heures utilisées.

Les trois types d'offres sont disponibles pour des durées d'un an et de trois ans.

Dimensionnez les nœuds réservés flexibles

Lorsque vous achetez un nœud réservé, vous devez notamment spécifier le type de nœud, par exemple db.r6g.xlarge. Pour plus d'informations sur les types de nœuds, consultez la section Tarification de MemoryDB.

Si vous avez un nœud et que vous devez le dimensionner pour augmenter sa capacité, votre nœud réservé est automatiquement appliqué à votre nœud redimensionné. En d'autres termes, vos nœuds réservés sont automatiquement appliqués à l'utilisation de n'importe quelle taille dans la même famille de nœuds. Des nœuds réservés de taille flexible sont disponibles pour les nœuds de la même AWS région. Les nœuds réservés dont la taille est flexible ne peuvent être redimensionnés que dans leurs familles de nœuds. Par exemple, un nœud réservé pour un fichier db.r6g.xlarge peut s'appliquer à un fichier db.r6g.2xlarge, mais pas à un fichier db.r6gd.large, car db.r6g et db.r6gd sont des familles de nœuds différentes.

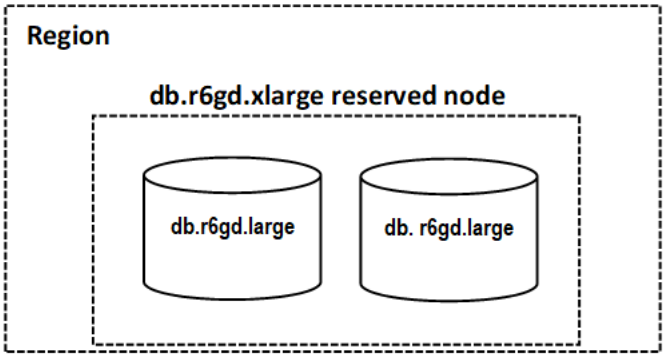
La flexibilité de taille signifie que vous pouvez passer librement d'une configuration à l'autre au sein d'une même famille de nœuds. Par exemple, vous pouvez passer d'un nœud réservé r6g.xlarge (8 unités normalisées) à deux nœuds réservés r6g.large (8 unités normalisées) (2*4 = 8 unités normalisées) dans la même région sans frais supplémentaires. AWS

Vous pouvez comparer l'utilisation de différentes tailles de nœuds réservés en utilisant des unités normalisées. Par exemple, une unité d'utilisation sur deux nœuds db.r6g.4xlarge équivaut à 16 unités d'utilisation normalisées sur un nœud db.r6g.large. Le tableau suivant indique le nombre d'unités normalisées pour chaque taille de nœud :

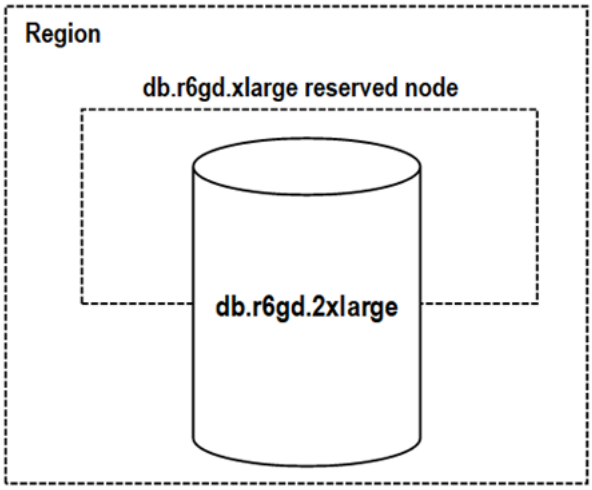
Taille de nœud	Unités normalisées
petit	1

Taille de nœud	Unités normalisées
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
6xlarge	48
8xlarge	64
10xlarge	80
12xlarge	96
16xlarge	128

Par exemple, vous achetez un nœud réservé db.r6gd.xlarge et deux nœuds réservés db.r6gd.large sont actifs sur votre compte dans la même région. AWS Dans ce cas, l'avantage de facturation est intégralement appliqué aux deux nœuds.



Sinon, si une instance db.r6gd.2xlarge est exécutée sur votre compte dans la même AWS région, l'avantage de facturation est appliqué à 50 % de l'utilisation du nœud réservé.



Supprimer un nœud réservé

Les conditions d'un nœud réservé impliquent un engagement d'un an ou de trois ans. Vous ne pouvez pas annuler un nœud réservé. Toutefois, vous pouvez supprimer un nœud couvert par une réduction sur les nœuds réservés. Le processus de suppression d'un nœud couvert par une réduction sur les nœuds réservés est le même que pour tout autre nœud.

Si vous supprimez un nœud couvert par une réduction sur les nœuds réservés, vous pouvez en lancer un autre avec des spécifications compatibles. Dans ce cas, vous conservez le tarif réduit jusqu'à la fin de la période de réservation (d'un ou de trois ans).

Utilisation de nœuds réservés

Vous pouvez utiliser les API AWS Management Console AWS Command Line Interface, the et MemoryDB pour travailler avec des nœuds réservés.

Console

Pour obtenir des prix et des informations sur les offres de nœuds réservés disponibles

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation, sélectionnez Reserved nodes.
3. Choisissez Acheter des nœuds réservés.
4. Pour Type de nœud, choisissez le type de nœud que vous souhaitez déployer.
5. Pour Quantité, choisissez le nombre de nœuds que vous souhaitez déployer.

6. Pour Term, choisissez la durée pendant laquelle vous souhaitez que le nœud de base de données soit réservé.
7. Pour Type d'offre, choisissez le type d'offre.

Après avoir effectué ces sélections, vous pouvez voir les informations tarifaires sous Résumé de la réservation.

> **Important**
>
> Choisissez Annuler pour éviter d'acheter ces nœuds réservés et d'encourir des frais.

Une fois que vous avez obtenu des informations sur les offres de nœuds réservés disponibles, vous pouvez les utiliser pour acheter une offre, comme indiqué dans la procédure suivante :

Pour acheter un nœud réservé

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation, sélectionnez Reserved nodes.
3. Choisissez Acheter des nœuds réservés.
4. Pour Type de nœud, choisissez le type de nœud que vous souhaitez déployer.
5. Pour Quantité, choisissez le nombre de nœuds que vous souhaitez déployer.
6. Pour Term, choisissez la durée pendant laquelle vous souhaitez que le nœud de base de données soit réservé.
7. Pour Type d'offre, choisissez le type d'offre.
8. (Facultatif) Vous pouvez attribuer votre propre identifiant aux nœuds réservés que vous achetez pour vous aider à les suivre. Dans le champ Numéro de réservation, saisissez un identifiant pour votre nœud réservé.

   Après avoir effectué ces sélections, vous pouvez voir les informations tarifaires sous Résumé de la réservation.
9. Choisissez Acheter des nœuds réservés.
10. Vos nœuds réservés sont achetés, puis affichés dans la liste des nœuds réservés.

Pour obtenir des informations sur les nœuds réservés à votre AWS compte

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation, sélectionnez Reserved nodes.
3. Les nœuds réservés à votre compte apparaissent. Pour obtenir des informations détaillées sur un nœud réservé en particulier, sélectionnez ce nœud dans la liste. Vous pouvez ensuite consulter des informations détaillées sur ce nœud dans le détail.

AWS Command Line Interface

L'`describe-reserved-nodes-offerings`exemple suivant renvoie les détails des offres de nœuds réservés.

```
aws memorydb describe-reserved-nodes-offerings
```

Cela produit un résultat similaire à ce qui suit :

```
{
    "ReservedNodesOfferings": [
        {
            "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
            "NodeType": "db.xxx.large",
            "Duration": 94608000,
            "FixedPrice": $xxx.xx,
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": $xx.xx,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        }
    ]
}
```

Vous pouvez également transmettre les paramètres suivants pour limiter l'étendue de ce qui est renvoyé :

- `--reserved-nodes-offering-id` – L'identifiant de l'offre que vous voulez acheter.
- `--node-type`— La valeur du filtre du type de nœud. Utilisez ce paramètre pour afficher uniquement les réservations correspondant au type de nœud spécifié.
- `--duration`— La valeur du filtre de durée, spécifiée en années ou en secondes. Utilisez ce paramètre pour afficher uniquement les réservations pour cette durée.
- `--offering-type`— Utilisez ce paramètre pour afficher uniquement les offres disponibles correspondant au type d'offre spécifié.

Une fois que vous avez obtenu des informations sur les offres de nœuds réservés disponibles, vous pouvez les utiliser pour acheter une offre.

L'`purchase-reserved-nodes-offering`exemple suivant achète de nouveaux nœuds réservés.

Pour Linux, macOS ou Unix :

```
aws memorydb purchase-reserved-nodes-offering \

    --reserved-nodes-offering-id 0193cc9d-7037-4d49-b332-d5e984f1d8ca \
    --reservation-id reservation \
    --node-count 2
```

Pour Windows :

```
aws memorydb purchase-reserved-nodes-offering ^
    --reserved-nodes-offering-id 0193cc9d-7037-4d49-b332-d5e984f1d8ca ^
    --reservation-id MyReservation
```

- `--reserved-nodes-offering-id`représente le nom des nœuds réservés proposant l'achat.
- `--reservation-id`est un identifiant spécifié par le client pour suivre cette réservation.

> **Note**
>
> Le numéro de réservation est un identifiant unique spécifié par le client pour suivre cette réservation. Si ce paramètre n'est pas spécifié, MemoryDB génère automatiquement un identifiant pour la réservation.

- `--node-count`est le nombre de nœuds à réserver. La valeur par défaut est 1.

Cela produit un résultat similaire à ce qui suit :

```
{
    "ReservedNode": {
        "ReservationId": "reservation",
        "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
        "NodeType": "db.xxx.large",
        "StartTime": 1671173133.982,
        "Duration": 94608000,
        "FixedPrice": $xxx.xx,
        "NodeCount": 2,
        "OfferingType": "Partial Upfront",
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": $xx.xx,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxx:reservednode/reservation"
    }
}
```

Après avoir acheté des nœuds réservés, vous pouvez obtenir des informations sur vos nœuds réservés.

L'`describe-reserved-nodes`exemple suivant renvoie des informations sur les nœuds réservés pour ce compte.

```
aws memorydb describe-reserved-nodes
```

Cela produit un résultat similaire à ce qui suit :

```
{
    "ReservedNodes": [
        {
            "ReservationId": "ri-2022-12-16-00-28-40-600",
            "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
            "NodeType": "db.xxx.large",
            "StartTime": 1671150737.969,
```

```
            "Duration": 94608000,
            "FixedPrice": $xxx.xx,
            "NodeCount": 1,
            "OfferingType": "Partial Upfront",
            "State": "active",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": $xx.xx,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxx:reservednode/
ri-2022-12-16-00-28-40-600"
        }
    ]
}
```

Vous pouvez également transmettre les paramètres suivants pour limiter l'étendue de ce qui est renvoyé :

- `--reservation-id`— Vous pouvez attribuer votre propre identifiant aux nœuds réservés que vous achetez pour faciliter leur suivi.
- `--reserved-nodes-offering-id`— La valeur du filtre de l'identifiant de l'offre. Utilisez ce paramètre pour afficher uniquement les réservations achetées correspondant à l'identifiant d'offre spécifié.
- `--node-type`— La valeur du filtre du type de nœud. Utilisez ce paramètre pour afficher uniquement les réservations correspondant au type de nœud spécifié.
- `--duration`— La valeur du filtre de durée, spécifiée en années ou en secondes. Utilisez ce paramètre pour afficher uniquement les réservations pour cette durée.
- `--offering-type`— Utilisez ce paramètre pour afficher uniquement les offres disponibles correspondant au type d'offre spécifié.

API MemoryDB

Les exemples suivants montrent comment utiliser l'API de requête MemoryDB pour les nœuds réservés :

DescribeReservedNodesOfferings

Renvoie les détails des offres de nœuds réservés.

```
https://memorydb.us-west-2.amazonaws.com/
    ?Action=DescribeReservedNodesOfferings
    &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
 &"Duration": 94608000,
    &NodeType="db.r6g.large"
    &OfferingType="Partial Upfront"
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Les paramètres suivants limitent l'étendue de ce qui est renvoyé :

- `ReservedNodesOfferingId`représente le nom des nœuds réservés proposant l'achat.
- `Duration`— La valeur du filtre de durée, spécifiée en années ou en secondes. Utilisez ce paramètre pour afficher uniquement les réservations pour cette durée.
- `NodeType`— La valeur du filtre du type de nœud. Utilisez ce paramètre pour afficher uniquement les offres correspondant au type de nœud spécifié.
- `OfferingType`— Utilisez ce paramètre pour afficher uniquement les offres disponibles correspondant au type d'offre spécifié.

Une fois que vous avez obtenu des informations sur les offres de nœuds réservés disponibles, vous pouvez les utiliser pour acheter une offre.

PurchaseReservedNodesOffering

Vous permet d'acheter une offre de nœuds réservés.

```
https://memorydb.us-west-2.amazonaws.com/
    ?Action=PurchaseReservedCacheNodesOffering
    &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
    &ReservationID=myreservationID
    &NodeCount=1
    &Version=2021-01-01
    &SignatureVersion=4
```

```
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

- `ReservedNodesOfferingId`représente le nom des nœuds réservés proposant l'achat.
- `ReservationID`Dest un identifiant spécifié par le client pour suivre cette réservation.

> **Note**
>
> Le numéro de réservation est un identifiant unique spécifié par le client pour suivre cette réservation. Si ce paramètre n'est pas spécifié, MemoryDB génère automatiquement un identifiant pour la réservation.

- `NodeCount`est le nombre de nœuds à réserver. La valeur par défaut est 1.

Après avoir acheté des nœuds réservés, vous pouvez obtenir des informations sur vos nœuds réservés.

DescribeReservedNodes

Renvoie des informations sur les nœuds réservés pour ce compte.

```
https://memorydb.us-west-2.amazonaws.com/
 ?Action=DescribeReservedNodes
 &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
 &ReservationID=myreservationID
 &NodeType="db.r6g.large"
 &Duration=94608000
 &OfferingType="Partial Upfront"
 &Version=2021-01-01
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20141201T220302Z
 &X-Amz-Algorithm
 &X-Amz-SignedHeaders=Host
 &X-Amz-Expires=20141201T220302Z
 &X-Amz-Credential=<credential>
```

```
&X-Amz-Signature=<signature>
```

Les paramètres suivants limitent l'étendue de ce qui est renvoyé :

- `ReservedNodesOfferingId`représente le nom du nœud réservé.
- `ReservationID`— Vous pouvez attribuer votre propre identifiant aux nœuds réservés que vous achetez pour faciliter leur suivi.
- `NodeType`— La valeur du filtre du type de nœud. Utilisez ce paramètre pour afficher uniquement les réservations correspondant au type de nœud spécifié.
- `Duration`— La valeur du filtre de durée, spécifiée en années ou en secondes. Utilisez ce paramètre pour afficher uniquement les réservations pour cette durée.
- `OfferingType`— Utilisez ce paramètre pour afficher uniquement les offres disponibles correspondant au type d'offre spécifié.

Consulter la facturation de vos nœuds réservés

Vous pouvez consulter la facturation de vos nœuds réservés dans le tableau de bord de facturation du AWS Management Console.

Pour consulter la facturation des nœuds réservés

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Sur le bouton Rechercher situé en haut de la console, sélectionnez Facturation.
3. Choisissez Bills dans la partie gauche du tableau de bord.
4. Sous Frais AWS de service, développez MemoryDB.
5. Développez la AWS région dans laquelle se trouvent vos nœuds réservés, par exemple USA Est (Virginie du Nord).

Vos nœuds réservés et leurs frais horaires pour le mois en cours sont indiqués sous Instances CreateCluster réservées Amazon MemoryDB.

Amazon MemoryDB CreateCluster Reserved Instances		
AmazonMemoryDB, db.r6g.large reserved instance applied	81.000 Hrs	
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	324.000 Hrs	
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	162.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6g.large instance	1,488.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6gd.2xlarge instance	744.000 Hrs	
USD 0 hourly fee per AmazonMemoryDB, db.r6g.4xlarge instance	744.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6gd.xlarge instance	744.000 Hrs	
USD 1 hourly fee per AmazonMemoryDB, db.r6gd.4xlarge instance	2,976.000 Hrs	

Remplacement de nœuds

MemoryDB met fréquemment à niveau son parc à l'aide de correctifs et de mises à niveau, généralement de manière fluide. Cependant, nous devons parfois relancer vos nœuds MemoryDB pour appliquer les mises à jour obligatoires du système d'exploitation à l'hôte sous-jacent. Ces remplacements sont obligatoires pour appliquer des mises à niveau qui renforcent la sécurité, la fiabilité et les performances opérationnelles.

Vous pouvez gérer ces remplacements vous-même à tout moment avant le créneau planifié de remplacement des nœuds. Lorsque vous gérez vous-même un remplacement, votre instance reçoit la mise à jour du système d'exploitation quand vous relancez le nœud. Le remplacement planifié du nœud est alors annulé. Il est possible que vous receviez encore des alertes indiquant que le remplacement du nœud aura lieu. Si vous avez déjà limité manuellement le besoin de maintenance, vous pouvez ignorer ces alertes.

> **Note**
>
> Les nœuds de remplacement générés automatiquement par MemoryDB peuvent avoir des adresses IP différentes. Il vous incombe de vérifier la configuration de votre application pour vous assurer que vos nœuds sont associés aux adresses IP appropriées.

La liste suivante identifie les actions que vous pouvez effectuer lorsque MemoryDB planifie le remplacement de l'un de vos nœuds :

Options de remplacement des nœuds MemoryDB

- Ne rien faire — Si vous ne faites rien, MemoryDB remplace le nœud comme prévu.

  Si le nœud est membre d'un cluster multi-AZ, MemoryDB améliore la disponibilité lors des correctifs, des mises à jour et des autres remplacements de nœuds liés à la maintenance.

  Le remplacement est terminé pendant que le cluster traite les demandes d'écriture entrantes.
- Modifiez votre fenêtre de maintenance — Pour les événements de maintenance planifiés, vous recevez un e-mail ou une notification de MemoryDB. Dans ce cas, si vous changez votre fenêtre de maintenance avant le créneau de remplacement planifié, votre nœud est désormais remplacé au nouvel horaire. Pour plus d'informations, consultez Modification d'un cluster MemoryDB.

> **Note**
>
> La possibilité de modifier votre fenêtre de remplacement en déplaçant votre fenêtre de maintenance n'est disponible que lorsque la notification MemoryDB inclut une fenêtre de maintenance. Si la notification ne comporte pas de fenêtre de maintenance, vous ne pouvez pas modifier votre fenêtre de remplacement.

Supposons par exemple que nous sommes le jeudi 9 novembre, qu'il est 15 h 00 et que la prochaine fenêtre de maintenance est vendredi 10 novembre à 17 h 00. Voici 3 scénarios avec leurs résultats :

- Vous reportez votre fenêtre de maintenance au vendredi à 16 h 00 (après la date et l'heure actuelles et avant la prochaine fenêtre de maintenance prévue). Le nœud est remplacé le vendredi 10 novembre à 16 h 00.
- Vous reportez votre fenêtre de maintenance au samedi à 16 h 00 (après la date et l'heure actuelles et après la prochaine fenêtre de maintenance prévue). Le nœud est remplacé le samedi 11 novembre à 16 h 00.
- Vous modifiez votre fenêtre de maintenance au mercredi à 16 h, plus tôt dans la semaine que la date et l'heure actuelles. Le nœud est remplacé le mercredi 15 novembre à 16 h 00.

Pour obtenir des instructions, consultez Gestion de la maintenance.

Gestion des clusters

La plupart des opérations MemoryDB sont effectuées au niveau du cluster. Vous pouvez définir un cluster de avec un nombre spécifique de nœuds de et un groupe de paramètres du qui contrôle les propriétés de chaque nœud de Tous les nœuds de au sein d'un cluster sont conçus pour avoir le même type de nœud et les mêmes paramètres et les mêmes configurations du groupe de sécurité.

Chaque cluster doit avoir un identifiant de cluster. L'identifiant de cluster est un nom fourni par le client pour le cluster. Cet identifiant spécifie un cluster particulier lors de l'interaction avec l'API et AWS CLI les commandes MemoryDB. L'identifiant du cluster doit être unique pour ce client dans une AWS région.

Les clusters MemoryDB sont conçus pour être accessibles via une instance Amazon EC2. Vous ne pouvez lancer votre cluster MemoryDB que dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC, mais vous pouvez y accéder depuis l'extérieur. AWS Pour plus d’informations, consultez Accès aux ressources de MemoryDB depuis l'extérieur AWS.

Mise à niveau des données

Les clusters qui utilisent un type de nœud de la famille r6gd voient leurs données hiérarchisées entre la mémoire et le stockage SSD (Solid State Drive) local. La hiérarchisation des données offre une nouvelle option de rapport prix/performances pour les charges de travail Redis OSS en utilisant des disques SSD à moindre coût dans chaque nœud du cluster, en plus du stockage des données en mémoire. Comme pour les autres types de nœuds, les données écrites sur les nœuds r6gd sont stockées de manière durable dans un journal des transactions multi-AZ. La hiérarchisation des données est parfaitement adaptée aux charges de travail qui accèdent régulièrement jusqu'à 20 % de leur jeu de données, et pour les applications qui peuvent tolérer une latence supplémentaire lors de l'accès aux données sur SSD.

Sur les clusters dotés d'une hiérarchisation des données, MemoryDB surveille l'heure du dernier accès de chaque élément stocké. Lorsque la mémoire disponible (DRAM) est entièrement consommée, MemoryDB utilise un algorithme utilisé le moins récemment (LRU) pour déplacer automatiquement les éléments rarement consultés de la mémoire vers le SSD. Lorsque les données du SSD sont ultérieurement consultées, MemoryDB les replace automatiquement et de manière asynchrone en mémoire avant de traiter la demande. Si votre charge de travail n’accède régulièrement qu’à un sous-ensemble de ses données, la hiérarchisation des données est un moyen optimal de mettre à l’échelle votre capacité de manière rentable.

Notez que lors de l'utilisation de la hiérarchisation des données, les clés elles-mêmes restent toujours en mémoire, tandis que le principe du moins récemment utilisé (LRU, Least Recently Used) régit le placement des valeurs en mémoire plutôt que sur le disque. En général, nous recommandons que la taille de vos clés soit inférieure à celle de vos valeurs lorsque vous utilisez la hiérarchisation des données.

La hiérarchisation des données est conçue pour avoir un impact minimal sur les performances des charges de travail des applications. Par exemple, en supposant des valeurs de chaîne de 500 octets, vous pouvez généralement vous attendre à 450 microsecondes de latence supplémentaires pour les demandes de lecture de données stockées sur un SSD par rapport aux demandes de lecture de données en mémoire.

Avec la plus grande taille de nœud de hiérarchisation des données (db.r6gd.8xlarge), vous pouvez stocker jusqu'à 500 To dans un seul cluster de 500 nœuds (250 To si vous utilisez une réplique en lecture). Pour la hiérarchisation des données, MemoryDB réserve 19 % de la mémoire (DRAM) par nœud à des fins autres que les données. La hiérarchisation des données est compatible avec toutes

les commandes et structures de données Redis OSS prises en charge dans MemoryDB. Vous n'avez besoin d'aucune modification côté client pour utiliser cette fonction.

Rubriques

- Bonnes pratiques
- Limites
- Mise à niveau de tarification des données
- Surveillance
- Utilisation de la hiérarchisation des données
- Restauration des données d'un instantané vers des clusters avec la hiérarchisation des données activée

Bonnes pratiques

Nous recommandons les bonnes pratiques suivantes :

- La hiérarchisation des données est parfaitement adaptée aux charges de travail qui accèdent régulièrement jusqu'à 20 % de leur jeu de données, et pour les applications qui peuvent tolérer une latence supplémentaire lors de l'accès aux données sur SSD.
- Lors de l'utilisation de la capacité SSD disponible sur les nœuds hiérarchisés en fonction des données, nous recommandons que la taille de la valeur soit supérieure à celle de la clé. La taille de la valeur ne peut pas être supérieure à 128 Mo ; sinon, elle ne sera pas déplacée sur le disque. Lorsque des éléments sont déplacés entre DRAM et SSD, les clés restent toujours en mémoire et seules les valeurs sont déplacées vers le niveau SSD.

Limites

La hiérarchisation des données présente les limitations suivantes :

- Le type de nœud que vous utilisez doit appartenir à la famille r6gd, disponible dans les régions suivantes : `us-east-2`, `us-east-1`, `us-west-2`, `us-west-1`, `eu-west-1`, `eu-west-3`, `eu-central-1`, `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `ap-south-1`, `ca-central-1` et `sa-east-1`.
- Vous ne pouvez pas restaurer un instantané d'un cluster r6gd dans un autre cluster à moins que celui-ci n'utilise également r6gd.

- Vous ne pouvez pas exporter un instantané vers Amazon S3 pour les clusters de hiérarchisation des données.
- L'enregistrement sans fonction fork n'est pas prise en charge.
- La mise à l'échelle n'est pas prise en charge depuis un cluster de hiérarchisation de données (par exemple, un cluster utilisant un type de nœud r6gd) vers un cluster qui n'utilise pas la hiérarchisation des données (par exemple, un cluster utilisant un type de nœud r6g).
- La hiérarchisation des données prend uniquement en charge les politiques de mémoire maximale `volatile-lru`, `allkeys-lru` et `noeviction`.
- Les éléments de plus de 128 MiB ne sont pas déplacés vers le SSD.

Mise à niveau de tarification des données

Les nœuds R6gd ont une capacité totale (mémoire+SSD) 5 fois supérieure et peuvent vous aider à réaliser des économies de stockage de plus de 60 % lorsqu'ils fonctionnent à une utilisation maximale par rapport aux nœuds R6g (mémoire uniquement). Pour plus d'informations, consultez la section Tarification de MemoryDB.

Surveillance

MemoryDB propose des métriques conçues spécifiquement pour surveiller les clusters de performance qui utilisent la hiérarchisation des données. Pour surveiller le ratio entre les éléments en DRAM et en SSD, vous pouvez utiliser la `CurrItems` métrique à.. Métriques pour MemoryDB Vous pouvez calculer le pourcentage comme suit : `(CurrItems with Dimension: Tier = Memory * 100) / (CurrItems with no dimension filter)` Lorsque le pourcentage d'éléments en mémoire tombe en dessous de 5 %, nous vous recommandons d'en tenir compteDimensionnement des clusters MemoryDB.

Pour plus d'informations, voir Métriques pour les clusters MemoryDB qui utilisent la hiérarchisation des données à. Métriques pour MemoryDB

Utilisation de la hiérarchisation des données

Utilisation de la hiérarchisation des données à l'aide du AWS Management Console

Lorsque vous créez un cluster, vous utilisez la hiérarchisation des données en sélectionnant un type de nœud de la famille r6gd, tel que db.r6gd.xlarge. La sélection de ce type de nœud active automatiquement la hiérarchisation des données.

Pour plus d'informations sur la création des clusters , consultez Étape 1 : créer un cluster.

Activation de la hiérarchisation des données à l'aide du AWS CLI

Lorsque vous créez un cluster à l'aide de AWS CLI, vous utilisez la hiérarchisation des données en sélectionnant un type de nœud de la famille r6gd, tel que db.r6gd.xlarge, et en définissant le paramètre. `--data-tiering`

Vous ne pouvez pas désactiver la hiérarchisation des données lorsque vous sélectionnez un type de nœud dans la famille r6gd. Si vous définissez le paramètre `--no-data-tiering`, l'opération échouera.

Pour Linux, macOS ou Unix :

```
aws memorydb create-cluster \
   --cluster-name my-cluster \
   --node-type db.r6gd.xlarge \
   --acl-name my-acl \
   --subnet-group my-sg \
   --data-tiering
```

Pour Windows :

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6gd.xlarge ^
   --acl-name my-acl ^
   --subnet-group my-sg
   --data-tiering
```

Après avoir exécuté cette opération, une réponse similaire à ceci s'affiche :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
```

```
        "NodeType": "db.r6gd.xlarge",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "DataTiering":"true",
        "AutoMinorVersionUpgrade": true
    }
}
```

Restauration des données d'un instantané vers des clusters avec la hiérarchisation des données activée

Vous pouvez restaurer un instantané sur un nouveau cluster avec la hiérarchisation des données activée à l'aide de la (console), (AWS CLI) ou (API MemoryDB). Lorsque vous créez un cluster à l’aide de types de nœuds de la famille r6gd, la hiérarchisation des données est activée.

Restauration des données d'un instantané vers des clusters avec la hiérarchisation des données activée (console)

Pour restaurer un instantané sur un nouveau cluster avec la hiérarchisation des données activée (console), suivez les étapes décrites dans Restauration à partir d'un instantané (console)

Notez que pour activer la hiérarchisation des données, vous devez sélectionner un type de nœud de la famille r6gd.

Restauration des données d'un instantané dans des clusters avec la hiérarchisation des données activée (AWS CLI)

Lors de la création d'un cluster à l'aide de AWS CLI, la hiérarchisation des données est utilisée par défaut en sélectionnant un type de nœud de la famille r6gd, tel que db.r6gd.xlarge, et en définissant le paramètre. `--data-tiering`

Vous ne pouvez pas désactiver la hiérarchisation des données lorsque vous sélectionnez un type de nœud dans la famille r6gd. Si vous définissez le paramètre `--no-data-tiering`, l'opération échouera.

Pour Linux, macOS ou Unix :

```
aws memorydb create-cluster \
   --cluster-name my-cluster \
   --node-type db.r6gd.xlarge \
   --acl-name my-acl \
   --subnet-group my-sg \
   --data-tiering \
   --snapshot-name my-snapshot
```

Pour Linux, macOS ou Unix :

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6gd.xlarge ^
   --acl-name my-acl ^
   --subnet-group my-sg ^
   --data-tiering ^
   --snapshot-name my-snapshot
```

Après avoir exécuté cette opération, une réponse similaire à ceci s'affiche :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6gd.xlarge",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
```

```
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "DataTiering": "true"
}
```

Préparation d'un cluster

Vous trouverez ci-dessous des instructions sur la création d'un cluster à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Chaque fois que vous créez un cluster, il est conseillé de procéder à des travaux préparatoires afin de ne pas avoir à le mettre à niveau ou à apporter des modifications immédiatement.

Rubriques

- Déterminer les exigences

Déterminer les exigences

Préparation

Connaître les réponses aux questions suivantes permet d'accélérer la création de votre cluster :

- Assurez-vous de créer un groupe de sous-réseaux dans le même VPC avant de commencer à créer un cluster. Vous pouvez également utiliser le groupe de sous-réseaux par défaut fourni. Pour plus d’informations, consultez Sous-réseaux et groupes de sous-réseaux.

  MemoryDB est conçu pour être accessible de l'intérieur à AWS l'aide d'Amazon EC2. Toutefois, si vous lancez un VPC basé sur Amazon VPC, vous pouvez fournir un accès depuis l'extérieur. AWS Pour plus d’informations, consultez Accès aux ressources de MemoryDB depuis l'extérieur AWS.
- Avez-vous besoin de personnaliser les valeurs des paramètres ?

  Si vous le faites, créez un groupe de paramètres personnalisé. Pour plus d’informations, consultez Création d'un groupe de paramètres.
- Devez-vous créer un groupe de sécurité VPC ?

  Pour plus d’informations, consultez Sécurité au sein de votre VPC.

- Comment avez-vous l'intention de mettre en œuvre la tolérance aux pannes ?

  Pour plus d’informations, consultez Atténuation des défaillances.

Rubriques

- Exigences relatives à la mémoire et au processeur
- Configuration du cluster MemoryDB
- Multiplexage E/S amélioré
- Exigences relatives au dimensionnement
- Exigences relatives à l'accès
- Région et zones de disponibilité

Exigences relatives à la mémoire et au processeur

L'élément de base de MemoryDB est le nœud. Les nœuds sont configurés en fragments pour former des clusters. En déterminant le type de nœud à utiliser pour votre cluster, tenez compte de la configuration de nœud du cluster et de la quantité de données à stocker.

Configuration du cluster MemoryDB

Les clusters MemoryDB sont composés de 1 à 500 partitions. Les données d'un cluster MemoryDB sont partitionnées entre les partitions du cluster. Votre application se connecte à un cluster MemoryDB à l'aide d'une adresse réseau appelée Endpoint. Outre les points de terminaison du nœud, le cluster MemoryDB lui-même possède un point de terminaison appelé point de terminaison du cluster. Votre application peut utiliser ce point de terminaison pour lire ou écrire dans le cluster, laissant à MemoryDB le soin de déterminer le nœud à partir duquel lire ou écrire.

Multiplexage E/S amélioré

Si vous utilisez Redis OSS version 7.0 ou supérieure, vous bénéficierez d'une accélération supplémentaire grâce au multiplexage d'E/S amélioré, dans le cadre duquel chaque thread d'E/S réseau dédié achemine les commandes de plusieurs clients vers le moteur Redis OSS, en tirant parti de la capacité de Redis OS à traiter efficacement les commandes par lots. Pour plus d'informations, voir Performances ultrarapides etthe section called “Types de nœuds pris en charge”.

Exigences relatives au dimensionnement

Tous les clusters peuvent être étendus à un type de nœud plus grand. Lorsque vous augmentez la taille d'un cluster MemoryDB, vous pouvez le faire en ligne pour qu'il reste disponible ou vous pouvez créer un nouveau cluster à partir d'un instantané et éviter que le nouveau cluster ne démarre à vide.

Pour plus d’informations, consultez Mise à l'échelle dans ce guide.

Exigences relatives à l'accès

De par leur conception, les clusters MemoryDB sont accessibles à partir d'instances Amazon EC2. L'accès réseau à un cluster MemoryDB est limité au compte qui a créé le cluster. Par conséquent, avant de pouvoir accéder à un cluster depuis une instance Amazon EC2, vous devez autoriser l'entrée dans le cluster. Pour plus d'informations, consultez Étape 2 : Autoriser l'accès au cluster dans ce manuel.

Région et zones de disponibilité

En localisant vos clusters MemoryDB dans une AWS région proche de votre application, vous pouvez réduire la latence. Si votre cluster dispose de plusieurs nœuds, la localisation de vos nœuds dans différentes zones de disponibilité peut réduire l'impact des échecs sur votre cluster.

Pour plus d’informations, consultez les ressources suivantes :

- Choix des régions et zones de disponibilité
- Atténuation des défaillances

Création d’un cluster

MemoryDB propose trois méthodes pour créer un cluster. Pour plus d’informations, consultez Étape 1 : créer un cluster.

Affichage des détails d'un cluster

Vous pouvez afficher des informations détaillées sur un ou plusieurs clusters à l'aide de la console MemoryDB ou de l'API AWS CLI MemoryDB.

Affichage des détails d'un cluster MemoryDB (console)

La procédure suivante explique comment afficher les détails d'un cluster MemoryDB à l'aide de la console MemoryDB.

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Pour voir les détails d'un cluster, cliquez sur le bouton radio situé à gauche du nom du cluster, puis choisissez Afficher les détails. Vous pouvez également cliquer directement sur le cluster pour afficher la page de détails du cluster.

   La page des détails du cluster affiche des informations sur le cluster, y compris le point de terminaison du cluster. Vous pouvez afficher plus de détails à l'aide des multiples onglets disponibles sur la page des détails du cluster.
3. Cliquez sur l'onglet Shards and nodes pour voir la liste des partitions du cluster et le nombre de nœuds dans chaque partition.
4. Pour afficher des informations spécifiques sur un nœud, développez le fragment dans le tableau ci-dessous. Vous pouvez également rechercher le fragment à l'aide du champ de recherche.

   Cela permet d'afficher des informations sur chaque nœud, notamment sa zone de disponibilité, ses slots/keyspaces et son statut.
5. Choisissez l'onglet Metrics pour surveiller leurs processus respectifs, tels que l'utilisation du processeur et l'utilisation du processeur du moteur. Pour plus d’informations, consultez Métriques pour MemoryDB.
6. Choisissez l'onglet Réseau et sécurité pour voir les détails du groupe de sous-réseaux et des groupes de sécurité.

   a. Dans le groupe de sous-réseaux, vous pouvez voir le nom du groupe de sous-réseaux, un lien vers le VPC auquel appartient le sous-réseau et le nom de ressource Amazon (ARN) du groupe de sous-réseaux.

   b. Dans Groupes de sécurité, vous pouvez voir l'ID, le nom et la description du groupe de sécurité.

7. Choisissez l'onglet Maintenance et capture d'écran pour voir les détails des paramètres de capture d'écran.

   a. Dans Snapshot, vous pouvez voir si les instantanés automatisés sont activés, quelle est la période de conservation des instantanés et quelle est la fenêtre des instantanés.

   b. Dans Snapshots, vous verrez une liste de tous les instantanés de ce cluster, y compris le nom, la taille, le nombre de partitions et le statut de l'instantané.

   Pour plus d'informations, consultez Instantané et restauration .

8. Choisissez l'onglet Maintenance et capture instantanée pour voir les détails de la fenêtre de maintenance, ainsi que les mises à jour de l'ACL, du repartage ou du service en attente. Pour plus d'informations, consultez Gestion de la maintenance.

9. Choisissez l'onglet Mises à jour de service pour voir les détails des mises à jour de service applicables à ce cluster. Pour plus d'informations, consultez Mises à jour du service dans MemoryDB.

10. Cliquez sur l'onglet Tags pour voir les détails de toutes les balises de ressources ou de répartition des coûts associées à ce cluster. Pour plus d'informations, consultez Marquage des instantanés.

Afficher les détails d'un cluster (AWS CLI)

Vous pouvez afficher les détails d'un cluster à l'aide de la AWS CLI `describe-clusters` commande. Si le paramètre `--cluster-name` n'est pas spécifié, les détails de plusieurs clusters jusqu'à `--max-results`, sont retournés. Si le paramètre `--cluster-name` est inclus, les détails du cluster spécifié sont retournés. Vous pouvez limiter le nombre d'enregistrements renvoyés avec le paramètre `--max-results`.

Le code suivant répertorie les détails de `my-cluster`.

```
aws memorydb describe-clusters --cluster-name my-cluster
```

Le code suivant affiche les détails de 25 clusters maximum.

```
aws memorydb describe-clusters --max-results 25
```

Example

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster \
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster ^
    --show-shard-details
```

La sortie JSON suivante montre la réponse :

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Description": "my cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": 1629230643.961,
                            "Endpoint": {
                                "Address": "my-cluster-0001-001.my-
cluster.abcdef.memorydb.us-east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
```

```
                            "Status": "available",
                            "CreateTime": 1629230644.025,
                            "Endpoint": {
                                "Address": "my-cluster-0001-002.my-
cluster.abcdef.memorydb.us-east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.abcdef.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "default",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:000000000:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "sat:06:30-sat:07:30",
            "SnapshotWindow": "04:00-05:00",
            "ACLName": "open-access",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true,
        }
```


Pour plus d'informations, consultez la rubrique dédiée AWS CLI à MemoryDB. [describe-clusters]

Afficher les détails d'un cluster (API MemoryDB)

Vous pouvez afficher les détails d'un cluster à l'aide de l'action d'API `DescribeClusters` MemoryDB. Si le paramètre `ClusterName` est inclus, les détails du cluster spécifié sont retournés. Si le paramètre `ClusterName` n'est pas spécifié, les détails de `MaxResults` clusters maximum (100 par défaut) sont retournés. La valeur de `MaxResults` ne peut pas être inférieure à 20 ou supérieure à 100.

Le code suivant répertorie les détails de `my-cluster`.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &ClusterName=my-cluster
   &Version=2021-01-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Le code suivant affiche les détails de 25 clusters maximum.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &MaxResults=25
   &Version=2021-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez la rubrique de référence de l'API MemoryDB.
`DescribeClusters`

Modification d'un cluster MemoryDB

Outre l'ajout ou la suppression de nœuds dans un cluster, il peut arriver que vous deviez apporter d'autres modifications à un cluster existant, par exemple en ajoutant un groupe de sécurité, en modifiant la fenêtre de maintenance ou un groupe de paramètres.

Nous vous conseillons que votre créneau de maintenance soit défini au moment où l'utilisation est la plus faible. Donc une modification peut s'avérer nécessaire de temps en temps.

Lorsque vous modifiez les paramètres d'un cluster, la modification est immédiatement appliquée au cluster. C'est vrai si vous changez le groupe de paramètres même du cluster ou une valeur de paramètre dans le groupe de paramètres du cluster.

Vous pouvez également mettre à jour la version du moteur de vos clusters. Par exemple, vous pouvez sélectionner une nouvelle version mineure du moteur et MemoryDB commencera immédiatement à mettre à jour votre cluster.

À l'aide du AWS Management Console

Pour modifier un cluster

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région dans laquelle se trouve le cluster que vous souhaitez modifier.
3. Dans le menu de navigation de gauche, accédez à Clusters. Dans Détails des clusters, sélectionnez le cluster à l'aide du bouton radio et accédez à Actions, puis à Modifier.
4. La page Modifier apparaît.
5. Dans la fenêtre Modifier, apportez les modifications souhaitées. Les options incluent :

    - Description
    - Groupes de sous-réseaux
    - Groupes de sécurité VPC
    - Type de nœud

> **Note**
>
> Si le cluster utilise un type de nœud de la famille r6gd, vous ne pouvez choisir qu'une taille de nœud différente de celle de cette famille. Si vous choisissez un type de nœud de la famille r6gd, la hiérarchisation des données sera automatiquement activée. Pour plus d'informations, consultez Mise à niveau des données.

- Compatibilité des versions Redis OSS
- Activer les instantanés automatiques
- Période de conservation des instantanés
- Fenêtre de capture instantanée
- Fenêtre de maintenance
- Rubrique pour la notification SNS

6. Sélectionnez Enregistrer les modifications.

Vous pouvez également accéder à la page des détails du cluster et cliquer sur Modifier pour apporter des modifications au cluster. Si vous souhaitez modifier des sections spécifiques du cluster, vous pouvez accéder à l'onglet correspondant de la page des détails du cluster et cliquer sur Modifier.

À l'aide du AWS CLI

Vous pouvez modifier un cluster existant à l'aide de cette AWS CLI `update-cluster` opération. Pour modifier la valeur de configuration d'un cluster, spécifiez l'ID du cluster, le paramètre à modifier et la nouvelle valeur du paramètre. L'exemple suivant change le créneau de maintenance pour un cluster nommé `my-cluster` et applique la modification immédiatement.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
```

```
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Pour plus d'informations, consultez update-cluster dans la référence des AWS CLI commandes.

Utilisation de l'API MemoryDB

Vous pouvez modifier un cluster existant à l'aide de l'opération d'API UpdateClusterMemoryDB. Pour modifier la valeur de configuration d'un cluster, spécifiez l'ID du cluster, le paramètre à modifier et la nouvelle valeur du paramètre. L'exemple suivant change le créneau de maintenance pour un cluster nommé `my-cluster` et applique la modification immédiatement.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &ClusterName=my-cluster
    &PreferredMaintenanceWindow=sun:23:00-mon:02:00
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210802T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Ajouter/supprimer des nœuds d'un cluster

Vous pouvez ajouter ou supprimer des nœuds d'un cluster à l'aide de l'API AWS Management Console, de AWS CLI, ou de l'API MemoryDB.

À l'aide du AWS Management Console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans la liste des clusters, choisissez le nom du cluster auquel vous souhaitez ajouter ou supprimer un nœud.
3. Dans l'onglet Shards and nodes, sélectionnez Add/Delete nodes
4. Dans Nouveau nombre de nœuds, entrez le nombre de nœuds souhaité.
5. Choisissez Confirmer.

> **Important**
>
> Si vous définissez le nombre de nœuds sur 1, le mode multi-AZ ne sera plus activé. Vous pouvez également choisir d'activer le basculement automatique.

À l'aide du AWS CLI

1. Identifiez les noms des nœuds que vous souhaitez supprimer. Pour plus d'informations, consultez Affichage des détails d'un cluster.
2. Utilisez l'opération `update-cluster` de la CLI avec une liste des nœuds à supprimer, comme dans l'exemple suivant.

   Pour supprimer des nœuds d'un cluster à l'aide de l'interface de ligne de commande, utilisez la commande `update-cluster` avec les paramètres suivants :

   - `--cluster-name`ID du cluster dont vous souhaitez supprimer des nœuds.
   - `--replica-configuration`— Permet de définir le nombre de répliques :
     - `ReplicaCount`— Définissez cette propriété pour spécifier le nombre de nœuds de réplication que vous souhaitez.
   - `--region`Spécifie la AWS région du cluster dont vous souhaitez supprimer des nœuds.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --replica-configuration \
        ReplicaCount=1 \
    --region us-east-1
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=1 ^
    --region us-east-1
```

Pour plus d'informations, consultez les AWS CLI rubriques `update-cluster`.

Utilisation de l'API MemoryDB

Pour supprimer des nœuds à l'aide de l'API MemoryDB, appelez l'opération d'`UpdateCluster`API avec le nom du cluster et une liste des nœuds à supprimer, comme indiqué :

- `ClusterName`ID du cluster dont vous souhaitez supprimer des nœuds.
- `ReplicaConfiguration`— Permet de définir le nombre de répliques :
  - `ReplicaCount`— Définissez cette propriété pour spécifier le nombre de nœuds de réplication que vous souhaitez.
- `Region`Spécifie la AWS région du cluster dont vous souhaitez supprimer un nœud.

Pour plus d'informations, consultez UpdateCluster.

Accès à votre cluster

Vos instances MemoryDB sont conçues pour être accessibles via une instance Amazon EC2.

Vous pouvez accéder à votre nœud MemoryDB depuis une instance Amazon EC2 dans le même Amazon VPC. Ou bien, en utilisant le peering VPC, vous pouvez accéder à votre nœud MemoryDB depuis un Amazon EC2 situé dans un autre Amazon VPC.

Rubriques

- Accordez l'accès à votre cluster
- Accès aux ressources de MemoryDB depuis l'extérieur AWS

Accordez l'accès à votre cluster

Vous ne pouvez vous connecter à votre cluster MemoryDB qu'à partir d'une instance Amazon EC2 exécutée dans le même Amazon VPC. Dans ce cas, vous devez accorder l'accès au réseau au cluster.

Pour accorder l'accès réseau à un cluster, à partir d'un groupe de sécurité Amazon VPC

1. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans le volet de navigation de gauche, sous Réseau et sécurité, choisissez Security Groups.
3. Dans la liste des groupes de sécurité, choisissez le groupe de sécurité de votre Amazon VPC. À moins que vous n'ayez créé un groupe de sécurité pour l'utilisation de MemoryDB, ce groupe de sécurité sera nommé par défaut.
4. Choisissez l'onglet Entrant et effectuez les opérations suivantes :
    a. Choisissez Edit (Modifier).
    b. Choisissez Ajouter une règle.
    c. Dans la colonne Type, choisissez Règle TCP personnalisée.
    d. Dans la zone Port range, tapez le numéro de port de votre nœud de cluster de Ce numéro doit être le même que celui que vous avez spécifié lorsque vous avez lancé le cluster. Le port par défaut pour Redis OSS est **6379**.

e. Dans le champ Source, choisissez Anywhere dont la plage de ports est comprise (0.0.0.0/0) afin que toute instance Amazon EC2 que vous lancez au sein de votre Amazon VPC puisse se connecter à vos nœuds MemoryDB.

> **Important**
>
> L'ouverture du cluster MemoryDB à 0.0.0.0/0 n'expose pas le cluster à Internet car il ne possède aucune adresse IP publique et n'est donc pas accessible depuis l'extérieur du VPC. Toutefois, le groupe de sécurité par défaut peut être appliqué aux autres instances Amazon EC2 dans le compte du client et ces instances peuvent voir une adresse IP publique. Si ces instances exécutent un service sur le port par défaut, ce service peut être exposé accidentellement. Par conséquent, nous vous recommandons de créer un groupe de sécurité VPC qui sera utilisé exclusivement par MemoryDB. Pour plus d'informations, consultez Groupes de sécurité personnalisés.

f. Choisissez Enregistrer.

Lorsque vous lancez une instance Amazon EC2 dans votre Amazon VPC, cette instance pourra se connecter à votre cluster MemoryDB.

Accès aux ressources de MemoryDB depuis l'extérieur AWS

MemoryDB est un service conçu pour être utilisé en interne dans votre VPC. L'accès externe est déconseillé en raison de la latence du trafic Internet et des problèmes de sécurité. Toutefois, si un accès externe à MemoryDB est requis à des fins de test ou de développement, il peut être effectué via un VPN.

À l'aide du AWS Client VPN, vous autorisez l'accès externe à vos nœuds MemoryDB avec les avantages suivants :

- Accès restreint aux utilisateurs approuvés ou aux clés d'authentification
- Trafic crypté entre le client VPN et le point de terminaison AWS VPN ;
- Accès limité à certains sous-réseaux ou nœuds
- Révocation facile de l'accès d'utilisateurs ou de clés d'authentification
- Audit des connexions

Les procédures suivantes montrent comment :

Rubriques

- Création d'une autorité de certification
- Configuration des composants VPN du AWS client
- Configuration du client VPN

Création d'une autorité de certification

Il est possible de créer une autorité de certification (CA) en utilisant différents outils ou techniques. Nous suggérons d'utiliser l'utilitaire easy-rsa, fourni par le projet OpenVPN. Quelle que soit l'option que vous choisissez, assurez-vous de garder les clés en sécurité. La procédure suivante télécharge les scripts easy-rsa, puis crée l'autorité de certification et les clés pour authentifier le premier client VPN :

- Pour créer les certificats initiaux, ouvrez un terminal et procédez comme suit :
  - `git clone` https://github.com/OpenVPN/easy-rsa
  - `cd easy-rsa`
  - `./easyrsa3/easyrsa init-pki`

- `./easyrsa3/easyrsa build-ca nopass`
- `./easyrsa3/easyrsa build-server-full server nopass`
- `./easyrsa3/easyrsa build-client-full client1.domain.tld nopass`

Un sous-répertoire pki contenant les certificats sera créé sous easy-rsa.

- Soumettez le certificat du serveur au gestionnaire de AWS certificats (ACM) :
  - Dans la console ACM, sélectionnez Certificate Manager.
  - Sélectionnez Import Certificate (Importer un certificat).
  - Entrez le certificat de clé publique disponible dans le fichier `easy-rsa/pki/issued/server.crt` dans le champ Corps du certificat.
  - Collez la clé privée disponible dans `easy-rsa/pki/private/server.key` dans le champ Clé privée du certificat. Assurez-vous de sélectionner toutes les lignes entre `BEGIN AND END PRIVATE KEY` (y compris les lignes `BEGIN` et `END`).
  - Collez la clé publique de l'autorité de certification disponible dans le fichier `easy-rsa/pki/ca.crt` dans le champ Chaîne de certificats.
  - Sélectionnez Vérifier et importer.
  - Sélectionnez Importer.

  Pour envoyer les certificats du serveur à ACM à l'aide de la AWS CLI, exécutez la commande suivante : `aws acm import-certificate --certificate fileb://easy-rsa/pki/issued/server.crt --private-key file://easy-rsa/pki/private/server.key --certificate-chain file://easy-rsa/pki/ca.crt --region region`

  Notez l'ARN du certificat pour un usage futur.

Configuration des composants VPN du AWS client

Utilisation de la AWS console

Sur la AWS console, sélectionnez Services, puis VPC.

Sous Virtual Private Network (Réseau privé virtuel), sélectionnez Client VPN Endpoints (Points de terminaison VPN client) et procédez comme suit :

Configuration des composants VPN du AWS Client

- Sélectionnez Create Client VPN Endpoint (Créer un point de terminaison VPN client).

- Spécifiez les options suivantes :
  - Client IPv4 CIDR (CIDR IPv4 client) : utilisez un réseau privé avec un masque réseau dont la plage est au moins de /22. Assurez-vous que le sous-réseau sélectionné n'est pas en conflit avec les adresses des réseaux VPC. Exemple : 10.0.0.0/22.
  - Dans Server certificate ARN (ARN du certificat de serveur), sélectionnez l'ARN du certificat que vous venez d'importer.
  - Sélectionnez Use mutual authentication (Utiliser l'authentification mutuelle).
  - Dans Client certificate ARN (ARN du certificat de client), sélectionnez l'ARN du certificat que vous venez d'importer.
  - Sélectionnez Create Client VPN Endpoint (Créer un point de terminaison VPN client).

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 create-client-vpn-endpoint --client-cidr-block
"10.0.0.0/22" --server-certificate-arn arn:aws:acm:us-
east-1:012345678912:certificate/0123abcd-ab12-01a0-123a-123456abcdef --
authentication-options Type=certificate-
authentication,,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:
east-1:012345678912:certificate/123abcd-ab12-01a0-123a-123456abcdef} --
connection-log-options Enabled=false
```

Exemple de sortie :

```
"ClientVpnEndpointId": "cvpn-endpoint-0123456789abcdefg",
"Status": { "Code": "pending-associate" }, "DnsName": "cvpn-
endpoint-0123456789abcdefg.prod.clientvpn.us-east-1.amazonaws.com" }
```

Association des réseaux cibles au point de terminaison VPN

- Sélectionnez le nouveau point de terminaison VPN, puis sélectionnez l'onglet Associations.
- Sélectionnez Associate (Associer), puis spécifiez les options suivantes.
  - VPC : sélectionnez le VPC du cluster MemoryDB.
  - Sélectionnez l'un des réseaux du cluster MemoryDB. En cas de doute, passez en revue les réseaux dans les groupes de sous-réseaux du tableau de bord MemoryDB.
  - Sélectionnez Associate (Associer). Si nécessaire, répétez les étapes pour les réseaux restants.

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 associate-client-vpn-target-network --client-vpn-endpoint-id cvpn-
endpoint-0123456789abcdefg --subnet-id subnet-0123456789abdcdef
```

Exemple de sortie :

```
"Status": { "Code": "associating" }, "AssociationId": "cvpn-
assoc-0123456789abdcdef" }
```

Examen du groupe de sécurité VPN

Le point de terminaison VPN adopte automatiquement le groupe de sécurité par défaut du VPC. Vérifiez les règles entrantes et sortantes et confirmez si le groupe de sécurité autorise le trafic du réseau VPN (défini dans les paramètres du point de terminaison VPN) vers les réseaux MemoryDB sur les ports de service (par défaut, 6379 pour Redis).

Si vous devez modifier le groupe de sécurité affecté au point de terminaison VPN, procédez comme suit :

- Sélectionnez le groupe de sécurité en cours.
- Sélectionnez Apply Security Group (Appliquer le groupe de sécurité).
- Sélectionnez le nouveau groupe de sécurité.

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 apply-security-groups-to-client-vpn-target-network --
client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefga  --vpc-id
vpc-0123456789abdcdef --security-group-ids sg-0123456789abdcdef
```

Exemple de sortie :

```
"SecurityGroupIds": [ "sg-0123456789abdcdef" ] }
```

> **Note**
>
> Le groupe de sécurité MemoryDB doit également autoriser le trafic provenant des clients VPN. Les adresses des clients sont masquées avec l'adresse du point de terminaison VPN,

selon le réseau VPC. Par conséquent, considérez le réseau VPC (et non le réseau des clients VPN) lors de la création de la règle entrante sur le groupe de sécurité MemoryDB.

Autorisation de l'accès VPN aux réseaux de destination

Dans l'onglet Authorization (Autorisation), sélectionnez Authorize Ingress (Autoriser l'entrée) et spécifiez les éléments suivants :

- Réseau de destination pour activer l'accès : utilisez 0.0.0.0/0 pour autoriser l'accès à n'importe quel réseau (y compris Internet) ou limitez les réseaux/hôtes MemoryDB.
- Sous Grant access to: (Accorder l'accès à :), sélectionnez Allow access to all users (Autoriser l'accès à tous les utilisateurs).
- Sélectionnez Add Authorization Rules (Ajouter des règles d'autorisation).

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 authorize-client-vpn-ingress --client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefg --target-network-cidr 0.0.0.0/0 --authorize-all-groups
```

Exemple de sortie :

```
{ "Status": { "Code": "authorizing" } }
```

Autorisation de l'accès à Internet à partir des clients VPN

Si vous avez besoin de naviguer sur Internet via le VPN, vous devez créer un routage supplémentaire. Sélectionnez l'onglet Route Table (Table de routage), puis sélectionnez Create Route (Créer un routage) :

- Destination du routage : 0.0.0.0/0
- Target VPC Subnet ID (ID de sous-réseau du VPC cible) : sélectionnez l'un des sous-réseaux associés ayant accès à Internet.
- Sélectionnez Create Route (Créer un routage).

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 create-client-vpn-route --client-vpn-endpoint-id cvpn-
endpoint-0123456789abcdefg --destination-cidr-block 0.0.0.0/0 --target-vpc-
subnet-id subnet-0123456789abdcdef
```

Exemple de sortie :

```
{ "Status": { "Code": "creating" } }
```

Configuration du client VPN

Sur le tableau de bord du AWS client VPN, sélectionnez le point de terminaison VPN récemment créé et sélectionnez Télécharger la configuration du client. Copiez le fichier de configuration, puis les fichiers `easy-rsa/pki/issued/client1.domain.tld.crt` et `easy-rsa/pki/private/client1.domain.tld.key`. Modifiez le fichier de configuration et modifiez ou ajoutez les paramètres suivants :

- cert : ajoutez une nouvelle ligne avec le paramètre cert pointant vers le fichier `client1.domain.tld.crt`. Utilisez le chemin complet du fichier. Exemple : `cert /home/user/.cert/client1.domain.tld.crt`
- key : ajoutez une nouvelle ligne avec le paramètre key pointant vers le fichier `client1.domain.tld.key`. Utilisez le chemin complet du fichier. Exemple : `key /home/user/.cert/client1.domain.tld.key`

Établissez la connexion VPN à l'aide de la commande : `sudo openvpn --config downloaded-client-config.ovpn`

Révocation de l'accès

Si vous devez invalider l'accès à partir d'une clé client particulière, la clé doit être révoquée dans l'autorité de certification. Soumettez ensuite la liste de révocation au AWS Client VPN.

Révocation de la clé avec easy-rsa :

- `cd easy-rsa`
- `./easyrsa3/easyrsa revoke client1.domain.tld`
- Entrez « yes » pour continuer, ou toute autre entrée pour abandonner.

```
Continue with revocation: `yes` ... * `./easyrsa3/easyrsa gen-crl
```

- Une liste de révocation de certificats (CRL) à jour a été créée. Fichier CRL : `/home/user/easy-rsa/pki/crl.pem`

Importation de la liste de révocation dans le VPN du AWS Client :

- Sur le AWS Management Console, sélectionnez Services, puis VPC.
- Sélectionnez Client VPN Endpoints (Points de terminaison VPN client).
- Sélectionnez le point de terminaison VPN client, puis sélectionnez Actions -> Import Client Certificate CRL (Importer une liste de révocation des certificats de client).
- Collez le contenu du fichier `crl.pem`.

À l'aide du AWS CLI

Exécutez la commande suivante :

```
aws ec2 import-client-vpn-client-certificate-revocation-list --certificate-revocation-list file://./easy-rsa/pki/crl.pem --client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefg
```

Exemple de sortie :

```
Example output: { "Return": true }
```

Recherche de points de terminaison de connexion

Votre application se connecte à votre cluster à l'aide du point de terminaison. Un point de terminaison est l'adresse unique d'un cluster. Utilisez le point de terminaison du cluster pour toutes les opérations.

Les sections suivantes vous aideront à découvrir le point de terminaison dont vous aurez besoin.

Trouver le point de terminaison d'un cluster MemoryDB ()AWS Management Console

Pour trouver le point de terminaison d'un cluster MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le panneau de navigation, choisissez Clusters.

   L'écran des clusters apparaîtra avec une liste de clusters. Choisissez le cluster auquel vous souhaitez vous connecter.
3. Pour trouver le point de terminaison du cluster, choisissez le nom du cluster (et non le bouton radio).
4. Le point de terminaison du cluster s'affiche sous Détails du cluster. Pour le copier, choisissez l'icône de copie à gauche du point de terminaison.

Trouver le point de terminaison d'un cluster MemoryDB (CLI)AWS

Vous pouvez utiliser la `describe-clusters` commande pour découvrir le point de terminaison d'un cluster. La commande renvoie le point de terminaison du cluster.

L'opération suivante récupère le point de terminaison, qui dans cet exemple est représenté sous forme d'*`exemple`*, pour le cluster`mycluster`.

Elle renvoie la réponse JSON suivante :

```
aws memorydb describe-clusters \
  --cluster-name mycluster
```

Pour Windows :

```
aws memorydb describe-clusters ^
   --cluster-name mycluster
```

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
```

```
            "NumberOfShards": 1,
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.4",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:zzzexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Pour plus d'informations, consultez describe-clusters.

Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Vous pouvez utiliser l'API MemoryDB pour découvrir le point de terminaison d'un cluster.

Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Vous pouvez utiliser l'API MemoryDB pour découvrir le point de terminaison d'un cluster avec l'`DescribeClusters`action. L'action renvoie le point de terminaison du cluster.

L'opération suivante permet de récupérer le point de terminaison du cluster`mycluster`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=mycluster
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T192317Z
    &Version=2021-01-01
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeClusters.

Utilisation de partitions

Un shard est un ensemble de 1 à 6 nœuds. Vous pouvez créer un cluster avec un plus grand nombre de partitions et un nombre inférieur de répliques, pour un total de 500 nœuds par cluster. Cette configuration de cluster peut aller de 500 partitions et 0 répliques à 100 partitions et 4 répliques, soit le nombre maximum de répliques autorisées. Les données du cluster sont partitionnées entre les partitions du cluster. S'il y a plus d'un nœud dans une partition, la partition met en œuvre la réplication avec un nœud qui est le nœud principal en lecture/écriture et les autres nœuds sont des nœuds de réplica en lecture seule.

Lorsque vous créez un cluster MemoryDB à l'aide de AWS Management Console, vous spécifiez le nombre de partitions dans le cluster et le nombre de nœuds dans les partitions. Pour plus d'informations, consultez Création d'un cluster MemoryDB.

Chaque nœud de partition possède les mêmes spécifications de calcul, de stockage et de mémoire. L'API MemoryDB vous permet de contrôler les attributs à l'échelle du cluster, tels que le nombre de nœuds, les paramètres de sécurité et les fenêtres de maintenance du système.

Pour plus d'informations, consultez Repartage hors ligne et rééquilibrage des partitions pour MemoryDB et Repartage en ligne et rééquilibrage des partitions pour MemoryDB.

Trouver le nom d'un shard

Vous pouvez trouver le nom d'une partition à l'aide de l'API AWS Management Console, de AWS CLI ou de MemoryDB.

À l'aide du AWS Management Console

La procédure suivante utilise le AWS Management Console pour rechercher les noms de partition d'un cluster MemoryDB.

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Clusters.
3. Sous Nom, choisissez le cluster dont vous souhaitez rechercher les noms de partition.
4. Sous l'onglet Partitions et nœuds, consultez la liste des partitions sous Nom. Vous pouvez également développer chacun d'eux pour afficher les détails de leurs nœuds.

À l'aide du AWS CLI

Pour trouver les noms de partitions (partitions) pour les clusters MemoryDB, utilisez l' AWS CLI opération `describe-clusters` avec le paramètre facultatif suivant.

- **`--cluster-name`**—Paramètre facultatif qui, lorsqu'il est utilisé, limite la sortie aux détails du cluster spécifié. Si ce paramètre est omis, les détails d'un maximum de 100 clusters sont renvoyés.
- **`--show-shard-details`**: renvoie les détails des fragments, y compris leurs noms.

Cette commande renvoie les informations relatives à `my-cluster`.

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Elle renvoie la réponse JSON suivante :

Des sauts de ligne sont ajoutés pour faciliter la lecture.

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
```

```
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Utilisation de l'API MemoryDB

Pour trouver les identifiants de partition pour les clusters MemoryDB, utilisez l'opération API `DescribeClusters` avec le paramètre facultatif suivant.

- **`ClusterName`**—Paramètre facultatif qui, lorsqu'il est utilisé, limite la sortie aux détails du cluster spécifié. Si ce paramètre est omis, les détails d'un maximum de 100 clusters sont renvoyés.
- **`ShowShardDetails`**: renvoie les détails des fragments, y compris leurs noms.

Example

Cette commande renvoie les informations relatives à `my-cluster`.

Pour Linux, macOS ou Unix :

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &ClusterName=sample-cluster
   &ShowShardDetails=true
   &Version=2021-01-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Gestion de votre implémentation de MemoryDB

Dans cette section, vous trouverez des informations sur la façon de gérer les différents composants de votre implémentation de MemoryDB.

Rubriques

- Versions du moteur Redis OSS
- Mise en route avec JSON
- Marquer vos ressources MemoryDB
- Gestion de la maintenance
- Bonnes pratiques
- Comprendre la réplication MemoryDB
- Instantané et restauration
- Mise à l'échelle
- Configuration des paramètres de moteur à l'aide de groupes de paramètres
- Tutoriel : Configuration d'une fonction Lambda pour accéder à MemoryDB dans un Amazon VPC

Versions du moteur Redis OSS

Cette section couvre les versions du moteur Redis OSS prises en charge.

Rubriques

- MemoryDB version 7.1 (améliorée)
- MemoryDB version 7.0 (améliorée)
- MemoryDB version 6.2 (améliorée)
- Mise à niveau des versions de moteur

MemoryDB version 7.1 (améliorée)

La version 7.1 de MemoryDB ajoute la prise en charge des fonctionnalités de recherche vectorielle en version préliminaire pour certaines régions, ainsi que des corrections de bogues critiques et des améliorations de performances.

- Fonction de recherche vectorielle : La recherche vectorielle peut être utilisée avec les fonctionnalités existantes de MemoryDB. Les applications qui n'utilisent pas la recherche vectorielle ne seront pas affectées par sa présence. L'aperçu de la recherche vectorielle est disponible à partir de la version 7.1 de MemoryDB dans les régions suivantes : USA Est (Virginie du Nord et Ohio), USA Ouest (Oregon), UE (Irlande) et Asie-Pacifique (Tokyo). Consultez la documentation ici pour savoir comment activer l'aperçu de la recherche vectorielle et les fonctionnalités associées.

> **Note**
>
> La version 7.1 de MemoryDB est compatible avec Redis OSS v7.0. Pour plus d'informations sur la version 7.0 de Redis OSS, consultez les notes de mise à jour de Redis OSS 7.0 sur Redis OSS on. GitHub

MemoryDB version 7.0 (améliorée)

MemoryDB 7.0 apporte un certain nombre d'améliorations et prend en charge de nouvelles fonctionnalités :

- Fonctions Redis OSS : MemoryDB 7 ajoute la prise en charge des fonctions Redis OSS et fournit une expérience gérée permettant aux développeurs d'exécuter des scripts LUA avec la logique d'application stockée sur le cluster MemoryDB, sans que les clients n'aient à renvoyer les scripts au serveur à chaque connexion.
- Améliorations des ACL : MemoryDB 7 ajoute le support pour la prochaine version des listes de contrôle d'accès (ACL) Redis OSS. Avec MemoryDB OSS 7, les clients peuvent désormais spécifier plusieurs ensembles d'autorisations sur des clés ou des espaces de touches spécifiques dans Redis OSS.
- Sharded Pub/Sub : MemoryDB 7 permet d'exécuter la fonctionnalité Redis OSS Pub/Sub de manière fragmentée lors de l'exécution de MemoryDB en mode cluster activé (CME). Les fonctionnalités Redis OSS Pub/Sub permettent aux éditeurs d'envoyer des messages à n'importe quel nombre d'abonnés sur une chaîne. Avec Amazon MemoryDB OSS 7, les canaux sont liés à une partition dans le cluster MemoryDB, ce qui élimine le besoin de propager les informations des canaux entre les partitions. Cela se traduit par une meilleure évolutivité.
- Multiplexage d'E/S amélioré : la version 7 de MemoryDB OSS introduit un multiplexage d'E/S amélioré, qui fournit un débit accru et une latence réduite pour les charges de travail à haut débit

qui ont de nombreuses connexions client simultanées à un cluster MemoryDB. Par exemple, lorsque vous utilisez un cluster de nœuds r6g.4xlarge et que vous exécutez 5 200 clients simultanés, vous pouvez augmenter le débit jusqu'à 46 % (opérations de lecture et d'écriture par seconde) et réduire la latence P99 de 21 % par rapport à la version 6 de MemoryDB.

Pour plus d'informations sur la version 7.0 de Redis OSS, consultez les notes de mise à jour de Redis OSS 7.0 sur Redis OSS on. GitHub

MemoryDB version 6.2 (améliorée)

MemoryDB présente la prochaine version du moteur Redis OSS, qui inclut la prise en charge de la mise à niveau automatique des versionsAuthentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL), la mise en cache côté client et des améliorations opérationnelles significatives.

La version 6.2.6 du moteur Redis intègre également la prise en charge du format natif JSON (JavaScript Object Notation), un moyen simple et sans schéma d'encoder des ensembles de données complexes dans des clusters Redis OSS. Grâce au support JSON, vous pouvez tirer parti des performances et des API Redis OSS pour les applications qui fonctionnent via JSON. Pour plus d’informations, consultez Mise en route avec JSON. Une métrique `JsonBasedCmds` liée au JSON est également incluse CloudWatch pour surveiller l'utilisation de ce type de données. Pour plus d’informations, consultez Métriques pour MemoryDB.

Avec Redis OSS 6, MemoryDB proposera une version unique pour chaque version mineure de Redis OSS, plutôt que de proposer plusieurs versions de correctif. Ceci est conçu pour minimiser la confusion et l'ambiguïté liées au choix entre plusieurs versions mineures. MemoryDB gérera également automatiquement la version mineure et la version corrective de vos clusters en cours d'exécution, garantissant ainsi de meilleures performances et une sécurité renforcée. Cela sera géré via des canaux de notification standard aux clients via une campagne de mise à jour du service. Pour plus d’informations, consultez Mises à jour du service dans MemoryDB.

Si vous ne spécifiez pas la version du moteur lors de la création, MemoryDB sélectionnera automatiquement la version Redis OSS préférée pour vous. D'autre part, si vous spécifiez la version du moteur en utilisant`6.2`, MemoryDB invoquera automatiquement la version de correctif préférée de Redis OSS 6.2 disponible.

Par exemple, lorsque vous créez un cluster, vous définissez le `--engine-version` paramètre sur`6.2`. Le cluster sera lancé avec la version de correctif préférée actuellement disponible au moment de sa création. Toute demande contenant une valeur de version complète du moteur sera rejetée, une exception sera émise et le processus échouera.

Lors de l'appel de l'`DescribeEngineVersions`API, la valeur du `EngineVersion` paramètre sera définie sur 6.2 et la version complète du moteur sera renvoyée `EnginePatchVersion` sur le terrain.

Pour plus d'informations sur la version 6.2 de Redis OSS, consultez les notes de mise à jour de Redis 6.2 sur Redis OSS on. GitHub

Mise à niveau des versions de moteur

Par défaut, MemoryDB gère automatiquement la version du correctif de vos clusters en cours d'exécution via des mises à jour de service. Vous pouvez également désactiver la mise à niveau automatique des versions mineures si vous définissez la `AutoMinorVersionUpgrade` propriété de vos clusters sur false. Cependant, vous ne pouvez pas désactiver la mise à niveau automatique de la version des correctifs.

Vous pouvez contrôler si et quand le logiciel conforme au protocole qui alimente votre cluster est mis à niveau vers de nouvelles versions prises en charge par MemoryDB avant le début de la mise à niveau automatique. Ce niveau de contrôle permet de maintenir la compatibilité avec des versions spécifiques, de tester les nouvelles versions avec votre application avant le déploiement en production et de réaliser des mises à niveau en fonction de vos propres conditions et délais.

Vous pouvez lancer des mises à niveau de la version du moteur vers votre cluster de la manière suivante :

- En le mettant à jour et en spécifiant une nouvelle version du moteur. Pour plus d'informations, consultez Modification d'un cluster MemoryDB.
- Appliquer la mise à jour de service pour la version du moteur correspondante. Pour plus d'informations, consultez Mises à jour du service dans MemoryDB.

Notez ce qui suit :

- Vous pouvez mettre à niveau vers une nouvelle version de moteur, mais vous ne pouvez pas revenir à une version antérieure de moteur. Si vous souhaitez utiliser une version antérieure de moteur, vous devez supprimer le cluster existant et en créer un nouveau avec la version de moteur antérieure.
- Nous vous recommandons de procéder à une mise à niveau périodique vers la dernière version majeure, car la plupart des améliorations majeures ne sont pas rétroportées vers les versions plus anciennes. À mesure que MemoryDB étend la disponibilité à une nouvelle AWS région, MemoryDB prend en charge les deux `MAJOR.MINOR` versions les plus récentes à ce moment-là pour la

nouvelle région. Par exemple, si une nouvelle AWS région est lancée et que les dernières versions de `MAJOR.MINOR` MemoryDB sont 7.0 et 6.2, MemoryDB prendra en charge les versions 7.0 et 6.2 dans la nouvelle région. AWS Au fur et à mesure que de nouvelles `MAJOR.MINOR` versions de MemoryDB seront publiées, MemoryDB continuera de prendre en charge les nouvelles versions de MemoryDB. Pour en savoir plus sur le choix des régions pour MemoryDB, consultez. Régions et terminaux pris en charge

- La gestion de la version du moteur est conçue afin que vous ayez autant de contrôle que possible sur le déroulement de la correction. MemoryDB se réserve toutefois le droit de corriger votre cluster en votre nom dans le cas peu probable d'une faille de sécurité critique dans le système ou le logiciel.
- MemoryDB proposera une version unique pour chaque version mineure de Redis OSS, plutôt que de proposer plusieurs versions de correctif. Ceci est conçu pour minimiser la confusion et l'ambiguïté liées au choix entre plusieurs versions. MemoryDB gérera également automatiquement la version mineure et la version corrective de vos clusters en cours d'exécution, garantissant ainsi de meilleures performances et une sécurité renforcée. Cela sera géré via des canaux de notification standard aux clients via une campagne de mise à jour du service. Pour plus d'informations, consultez Mises à jour du service dans MemoryDB.
- Vous pouvez mettre à niveau la version de votre cluster avec un temps d'arrêt minimal. Le cluster est disponible pour la lecture pendant toute la mise à niveau et reste disponible pour l'écriture pendant la majeure partie de la mise à niveau, sauf durant l'opération de basculement, qui dure quelques secondes.
- Nous vous recommandons d'effectuer des mises à niveau du moteur pendant les périodes de faible trafic d'écriture entrant.

  Les clusters contenant plusieurs partitions sont traités et corrigés comme suit :

  - Une seule opération de mise à niveau est effectuée par partition à la fois.
  - Dans chaque partition, tous les réplicas sont traités avant le réplica principal. S'il y a moins de réplicas dans une partition, le réplica principal de cette partition peut être traité avant que le traitement des réplicas des autres partitions ne soit terminé.
  - Dans toutes les partitions, les nœuds principaux sont traités en séries. Un seul nœud principal est mis à niveau à la fois.

Rubriques

- Comment mettre à niveau les versions de moteur
- Résolution des mises à niveau bloquées du moteur Redis OSS

Comment mettre à niveau les versions de moteur

Vous initiez les mises à niveau de version de votre cluster en le modifiant à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB et en spécifiant une version du moteur plus récente. Pour plus d'informations, consultez les rubriques suivantes.

- À l'aide du AWS Management Console
- À l'aide du AWS CLI
- Utilisation de l'API MemoryDB

Résolution des mises à niveau bloquées du moteur Redis OSS

Comme indiqué dans le tableau suivant, votre opération de mise à niveau du moteur Redis OSS est bloquée si vous avez une opération de mise à l'échelle en attente.

Opérations en suspens	Opérations bloquées
Mise à l'échelle ascendante	Mise à niveau du moteur
Mise à niveau du moteur	Mise à niveau du moteur
Augmentation et mise à niveau du moteur	Mise à niveau du moteur
	Mise à niveau du moteur

Mise en route avec JSON

MemoryDB prend en charge le format natif JSON (JavaScript Object Notation), un moyen simple et sans schéma d'encoder des ensembles de données complexes dans des clusters Redis OSS. Vous pouvez stocker et accéder aux données de manière native à l'aide du format JSON (JavaScript Object Notation) dans les clusters Redis OSS et mettre à jour les données JSON stockées dans ces clusters, sans avoir à gérer de code personnalisé pour le sérialiser et le désérialiser.

En plus de tirer parti des API Redis OSS pour les applications qui fonctionnent via JSON, vous pouvez désormais récupérer et mettre à jour efficacement des parties spécifiques d'un document JSON sans avoir à manipuler l'objet dans son intégralité, ce qui peut améliorer les performances et

réduire les coûts. Vous pouvez également rechercher le contenu de votre document JSON à l'aide de la requête `JSONPath` de style Goessner.

Après avoir créé un cluster avec une version de moteur prise en charge, le type de données JSON et les commandes associées sont automatiquement disponibles. Ceci est compatible avec l'API et compatible RDB avec la version 2 du module RedisJSON, ce qui vous permet de migrer facilement les applications Redis OSS existantes basées sur JSON vers MemoryDB. Pour plus d'informations sur les commandes Redis OSS prises en charge, consultezCommandes prises en charge.

Une métrique liée au JSON `JsonBasedCmds` est intégrée CloudWatch pour surveiller l'utilisation de ce type de données. Pour plus d'informations, consultez Metrics for MemoryDB.

> **Note**
>
> Pour utiliser JSON, vous devez exécuter le moteur Redis OSS version 6.2.6 ou ultérieure.

Rubriques

- Présentation du type de données JSON Redis OSS
- Commandes prises en charge

Présentation du type de données JSON Redis OSS

MemoryDB prend en charge un certain nombre de commandes Redis OSS pour travailler avec le type de données JSON. Vous trouverez ci-dessous un aperçu du type de données JSON et une liste détaillée des commandes Redis OSS prises en charge.

Terminologie

Terme	Description
Document JSON	fait référence à la valeur d'une clé JSON Redis OSS
Valeur JSON	fait référence à un sous-ensemble d'un document JSON, y compris la racine qui représente l'intégralité du document. Une

Terme	Description
	valeur peut être un conteneur ou une entrée dans un conteneur
Élément JSON	équivalent à une valeur JSON

Norme JSON prise en charge

Le format JSON est conforme à la norme d'échange de données JSON RFC 7159 et ECMA-404. L'Unicode UTF-8 dans le texte JSON est pris en charge.

Élément racine

L'élément racine peut être de n'importe quel type de données JSON. Notez que dans la précédente RFC 4627, seuls les objets ou les tableaux étaient autorisés comme valeurs racine. Depuis la mise à jour vers la RFC 7159, la racine d'un document JSON peut être de n'importe quel type de données JSON.

Limite de taille du document

Les documents JSON sont stockés en interne dans un format optimisé pour un accès et une modification rapides. Ce format consomme généralement un peu plus de mémoire que la représentation sérialisée équivalente du même document. La consommation de mémoire par un seul document JSON est limitée à 64 Mo, ce qui correspond à la taille de la structure de données en mémoire, et non à la chaîne JSON. La quantité de mémoire consommée par un document JSON peut être inspectée à l'aide de la `JSON.DEBUG MEMORY` commande.

Listes ACL JSON

- Le type de données JSON est entièrement intégré à la fonctionnalité Redis Access Control Lists (ACL). À l'instar des catégories existantes par type de données (@string, @hash, etc.), une nouvelle catégorie @json est ajoutée pour simplifier la gestion de l'accès aux commandes et aux données JSON. Aucune autre commande Redis OSS existante n'appartient à la catégorie @json. Toutes les commandes JSON appliquent les restrictions et autorisations des keyspaces ou des commandes.
- Cinq catégories d'ACL Redis OSS existantes ont été mises à jour pour inclure les nouvelles commandes JSON : @read, @write, @fast, @slow et @admin. Le tableau ci-dessous indique le mappage des commandes JSON vers les catégories appropriées.

ACL

Commande JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		

Commande JSON	@read	@write	@fast	@slow	@admin
JSON.NUMMULTBY		y	y		
JSON.OBJKEYS	y		y		
JSON.OBJLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLEN	y		y		
JSON.STRLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

Limite de profondeur d'imbrication

Lorsqu'un objet ou un tableau JSON possède un élément qui est lui-même un autre objet ou tableau JSON, cet objet ou tableau intérieur est dit « imbriqué » dans l'objet ou le tableau extérieur. La limite maximale de la profondeur d'imbrication est de 128. Toute tentative de création d'un document contenant une profondeur d'imbrication supérieure à 128 sera rejetée avec une erreur.

Syntaxe de commande

La plupart des commandes nécessitent un nom de clé Redis OSS comme premier argument. Certaines commandes ont également un argument path. L'argument path prend par défaut la racine s'il est facultatif et non fourni.

Notation :

- Les arguments obligatoires sont placés entre crochets, par exemple <key>
- Les arguments facultatifs sont placés entre crochets, par exemple [path]
- Les arguments facultatifs supplémentaires sont indiqués par..., par exemple [json...]

Syntaxe de chemin

JSON-Redis OSS prend en charge deux types de syntaxes de chemin :

- Syntaxe améliorée — Suit la syntaxe JSONPath décrite par Goessner, comme indiqué dans le tableau ci-dessous. Nous avons réorganisé et modifié les descriptions dans le tableau pour plus de clarté.
- Syntaxe restreinte : possède des capacités de requête limitées.

> **Note**
>
> Les résultats de certaines commandes dépendent du type de syntaxe de chemin utilisé.

Si un chemin de requête commence par « $ », il utilise la syntaxe améliorée. Sinon, la syntaxe restreinte est utilisée.

Syntaxe améliorée

Symbole/Expression	Description
$	l'élément racine
. ou []	enfant opérateur
..	descente récursive

Symbole/Expression	Description
*	joker. Tous les éléments d'un objet ou d'un tableau.
[]	opérateur d'indice de tableau. L'index est basé sur 0.
[,]	opérateur syndical
[start:end:step]	opérateur de tranche de tableau
?()	applique une expression de filtre (script) au tableau ou à l'objet en cours
()	expression du filtre
@	utilisé dans les expressions de filtre faisant référence au nœud en cours de traitement
==	égal à, utilisé dans les expressions de filtre.
!=	différent de, utilisé dans les expressions de filtre.
>	supérieur à, utilisé dans les expressions de filtre.
>=	supérieur ou égal à, utilisé dans les expressio ns de filtre.
<	inférieur à, utilisé dans les expressions de filtre.
<=	inférieur ou égal à, utilisé dans les expressions de filtre.
&&	ET logique, utilisé pour combiner plusieurs expressions de filtre.

Symbole/Expression	Description
||	OR logique, utilisé pour combiner plusieurs expressions de filtre.

Exemples

Les exemples ci-dessous sont basés sur les exemples de données XML de Goessner, que nous avons modifiés en ajoutant des champs supplémentaires.

```
{ "store": {
    "book": [
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95,
        "in-stock": true,
        "sold": true
      },
      { "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99,
        "in-stock": false,
        "sold": true
      },
      { "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99,
        "in-stock": true,
        "sold": false
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99,
        "in-stock": false,
        "sold": false
      }
```

```
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95,
      "in-stock": true,
      "sold": false
    }
  }
}
```

Chemin	Description
$.store.book[*].author	les auteurs de tous les livres de la boutique
$..author	tous les auteurs
$.store.*	tous les membres de la boutique
$["store"].*	tous les membres de la boutique
$.store..price	le prix de tout ce qui se trouve dans le magasin
$..*	tous les membres récursifs de la structure JSON
$..book[*]	tous les livres
$..book[0]	le premier livre
$..book[-1]	le dernier livre
$..book[0:2]	les deux premiers livres
$..book[0,1]	les deux premiers livres
$..book[0:4]	livres de l'index 0 à 3 (l'index final n'est pas inclus)
$..book[0:4:2]	livres aux index 0, 2
$..book[?(@.isbn)]	tous les livres avec numéro ISBN

Chemin	Description
$..book[?(@.price<10)]	tous les livres à moins de 10$
'$..book[?(@.price < 10)]'	tous les livres moins chers que 10$. (Le chemin doit être entre guillemets s'il contient des espaces)
'$..book[?(@["price"] < 10)]'	tous les livres à moins de 10$
'$..book[?(@.["price"] < 10)]'	tous les livres à moins de 10$
$.. book [? (@.prix>=10&&@.prix<=100)]	tous les livres dans la fourchette de prix de 10$ à 100$, inclus
'$..book[?(@.price>=10 && @.price<=100)]'	tous les livres dont le prix varie de 10$ à 100$, inclus. (Le chemin doit être entre guillemets s'il contient des espaces)
$..book[?(@.sold==true||@.in-stock==false)]	tous les livres sont vendus ou en rupture de stock
'$..book[?(@.sold == true || @.in-stock == false)]'	tous les livres sont vendus ou sont en rupture de stock. (Le chemin doit être entre guillemets s'il contient des espaces)
'$.store.book[?(@.["category"] == "fiction")]'	tous les livres de la catégorie fiction
'$.store.book[?(@.["category"] != "fiction")]'	tous les livres dans les catégories non-ficti onnelles

Autres exemples d'expressions de filtre :

```
127.0.0.1:6379> JSON.SET k1 . '{"books": [{"price":5,"sold":true,"in-
stock":true,"title":"foo"}, {"price":15,"sold":false,"title":"abc"}]}'
OK
127.0.0.1:6379> JSON.GET k1 $.books[?(@.price>1&&@.price<20&&@.in-stock)]
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.price>1 && @.price<20 && @.in-stock)]'
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
```

```
127.0.0.1:6379> JSON.GET k1 '$.books[?((@.price>1 && @.price<20) && (@.sold==false))]'
"[{\"price\":15,\"sold\":false,\"title\":\"abc\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.title == "abc")]'
[{"price":15,"sold":false,"title":"abc"}]

127.0.0.1:6379> JSON.SET k2 . '[1,2,3,4,5]'
127.0.0.1:6379> JSON.GET k2 $.*.[?(@>2)]
"[3,4,5]"
127.0.0.1:6379> JSON.GET k2 '$.*.[?(@ > 2)]'
"[3,4,5]"

127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
OK
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
"[true,true]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
"[true,true]"
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
"[2,3,4]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
"[2,3,4]"
```

Syntaxe restreinte

Symbole/Expression	Description
. ou []	enfant opérateur
[]	opérateur d'indice de tableau. L'index est basé sur 0.

Exemples

Chemin	Description
.store.book[0].author	l'auteur du premier livre
.store.book[-1].author	l'auteur du dernier livre
.address.city	nom de la ville

Chemin	Description
["store"]["book"][0]["title"]	le titre du premier livre
["store"]["book"][-1]["title"]	le titre du dernier livre

> **Note**
>
> Tout le contenu de Goessner cité dans cette documentation est soumis à la licence Creative Commons.

Préfixes d'erreur courantes

Chaque message d'erreur possède un préfixe. Voici une liste des préfixes d'erreur courants :

Préfixe	Description
ERR	une erreur générale
LIMIT	erreur de dépassement de la limite de taille. Par exemple, la limite de taille du document ou la limite de profondeur d'imbrication ont été dépassées
NONEXISTENT	une clé ou un chemin n'existe pas
OUTOFBOUNDARIES	index de tableau hors limites
SYNTAXERR	erreur de syntaxe
WRONGTYPE	type de valeur incorrect

Métriques liées au JSON

Les métriques d'informations JSON suivantes sont fournies :

Infos	Description
json_total_memory_bytes	mémoire totale allouée aux objets JSON
json_num_documents	nombre total de documents dans Redis OSS

Pour interroger les métriques de base, exécutez la commande Redis OSS :

```
info json_core_metrics
```

Comment MemoryDB interagit avec JSON

Ce qui suit illustre comment MemoryDB interagit avec le type de données JSON.

Priorité des opérateurs

Lors de l'évaluation d'expressions conditionnelles pour le filtrage, les && sont prioritaires, puis les || sont évalués, comme c'est le cas dans la plupart des langages. Les opérations entre parenthèses seront exécutées en premier.

Comportement de la limite maximale d'imbrication des chemins

La limite maximale d'imbrication de chemins de MemoryDB est de 128. Ainsi, une valeur comme `$.a.b.c.d...` ne peut atteindre que 128 niveaux.

Traitement des valeurs numériques

Le JSON ne dispose pas de types de données distincts pour les nombres entiers et les nombres à virgule flottante. Ils sont tous appelés des nombres.

Lorsqu'un numéro JSON est reçu, il est stocké dans l'un des deux formats suivants. Si le nombre correspond à un entier signé de 64 bits, il est converti dans ce format ; sinon, il est stocké sous forme de chaîne. Les opérations arithmétiques sur deux nombres JSON (par exemple JSON.NUMINCRBY et JSON.NUMMULTBY) tentent de préserver autant de précision que possible. Si les deux opérandes et la valeur résultante correspondent à un entier signé de 64 bits, l'arithmétique des entiers est effectuée. Sinon, les opérandes d'entrée sont convertis en nombres à virgule flottante IEEE à double précision 64 bits, l'opération arithmétique est effectuée et le résultat est reconverti en chaîne.

Commandes arithmétiques `NUMINCRBY` et `NUMMULTBY` :

- Si les deux nombres sont des entiers et que le résultat est hors de la plage de int64, il deviendra automatiquement un nombre à virgule flottante à double précision.
- Si au moins l'un des nombres est un nombre à virgule flottante, le résultat sera un nombre à virgule flottante à double précision.
- Si le résultat dépasse la plage de deux, la commande renvoie une `OVERFLOW` erreur.

> **Note**
>
> Avant la version 6.2.6.R2 du moteur Redis OSS, lorsqu'un numéro JSON était reçu en entrée, il était converti en l'une des deux représentations binaires internes : un entier signé de 64 bits ou un nombre à virgule flottante IEEE à double précision de 64 bits. La chaîne de caractères d'origine et toute sa mise en forme ne sont pas retenues. Ainsi, lorsqu'un nombre est généré en sortie dans le cadre d'une réponse JSON, il est converti de la représentation binaire interne en une chaîne imprimable qui utilise des règles de formatage génériques. Ces règles peuvent entraîner la génération d'une chaîne différente de celle qui a été reçue.
>
> - Si les deux nombres sont des entiers et que le résultat est hors de la plage de `int64`, il devient automatiquement un nombre à virgule flottante IEEE à double précision de 64 bits.
> - Si au moins un des nombres est un nombre à virgule flottante, le résultat est un nombre à virgule flottante IEEE à double précision de 64 bits.
> - Si le résultat dépasse la plage du double IEEE de 64 bits, la commande renvoie une erreur `OVERFLOW`.

Pour une liste détaillée des commandes disponibles, consultez Commandes prises en charge.

Évaluation stricte de la syntaxe

MemoryDB n'autorise pas les chemins JSON dont la syntaxe est invalide, même si un sous-ensemble du chemin contient un chemin valide. Ceci afin de maintenir un comportement correct pour nos clients.

Commandes prises en charge

Les commandes JSON Redis OSS suivantes sont prises en charge :

Rubriques

- JSON.ARRAPPEND
- JSON.ARRINDEX
- JSON.ARRINSERT
- JSON.ARRLEN
- JSON.ARRPOP
- JSON.ARRTRIM
- JSON.CLEAR
- JSON.DEBUG
- JSON.DEL
- JSON.FORGET
- JSON.GET
- JSON.MGET
- JSON.NUMINCRBY
- JSON.NUMMULTBY
- JSON.OBJLEN
- JSON.OBJKEYS
- JSON.RESP
- JSON.SET
- JSON.STRAPPEND
- JSON.STRLEN
- JSON.TOGGLE
- JSON.TYPE

JSON.ARRAPPEND

Ajoutez une ou plusieurs valeurs aux valeurs du tableau sur le chemin.

Syntaxe

```
JSON.ARRAPPEND <key> <path> <json> [json ...]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- json (obligatoire) — Valeur JSON à ajouter au tableau

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la nouvelle longueur du tableau à chaque chemin.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.
- Erreur `SYNTAXERR` si l'un des arguments d'entrée json n'est pas une chaîne JSON valide.
- Erreur `NONEXISTENT` si le chemin n'existe pas.

Si le chemin est une syntaxe restreinte :

- Entier, la nouvelle longueur du tableau.
- Si plusieurs valeurs de tableau sont sélectionnées, la commande renvoie la nouvelle longueur du dernier tableau mis à jour.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.
- Erreur `SYNTAXERR` si l'un des arguments d'entrée json n'est pas une chaîne JSON valide.
- Erreur `NONEXISTENT` si le chemin n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 $[*] '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"a\",\"c\"],[\"a\",\"b\",\"c\"]]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 [-1] '"c"'
(integer) 3
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"],[\"a\",\"b\",\"c\"]]"
```

JSON.ARRINDEX

Recherchez la première occurrence d'une valeur JSON scalaire dans les tableaux situés sur le chemin.

- Les erreurs hors limites sont traitées en arrondissant l'index au début et à la fin du tableau.
- Si start > end, retourner -1 (non trouvé).

Syntaxe

```
JSON.ARRINDEX <key> <path> <json-scalar> [start [end]]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- json-scalar (obligatoire) — valeur scalaire à rechercher ; le scalaire JSON fait référence à des valeurs qui ne sont pas des objets ou des tableaux. C'est-à-dire que String, number, boolean et null sont des valeurs scalaires.
- start (facultatif) — index de départ inclus. La valeur par défaut est 0 si elle n'est pas fournie.
- end (facultatif) — index de fin, exclusif. La valeur par défaut est 0 s'il n'est pas fourni, ce qui signifie que le dernier élément est inclus. 0 ou -1 signifie que le dernier élément est inclus.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers. Chaque valeur est l'index de l'élément correspondant dans le tableau au niveau du chemin. La valeur est -1 si elle n'est pas trouvée.

- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.

Si le chemin est une syntaxe restreinte :

- Entier, l'index de l'élément correspondant, ou -1 si non trouvé.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 $[*] '"b"'
1) (integer) -1
2) (integer) -1
3) (integer) 1
4) (integer) 1
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 .children '"Tom"'
(integer) 2
```

JSON.ARRINSERT

Insérez une ou plusieurs valeurs dans les valeurs du tableau au niveau du chemin avant l'index.

Syntaxe

```
JSON.ARRINSERT <key> <path> <index> <json> [json ...]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- index (obligatoire) — index du tableau avant lequel les valeurs sont insérées.

- json (obligatoire) — Valeur JSON à ajouter au tableau

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la nouvelle longueur du tableau à chaque chemin.
- Si une valeur est un tableau vide, sa valeur de retour correspondante est nulle.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.
- Erreur `OUTOFBOUNDARIES` si l'argument index est hors limites.

Si le chemin est une syntaxe restreinte :

- Entier, la nouvelle longueur du tableau.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.
- Erreur `OUTOFBOUNDARIES` si l'argument index est hors limites.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 $[*] 0 '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"c\",\"a\"],[\"c\",\"a\",\"b\"]]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 . 0 '"c"'
(integer) 4
127.0.0.1:6379> JSON.GET k1
```

```
"[\"c\",[],[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRLEN

Obtenez la longueur des valeurs du tableau sur le chemin.

Syntaxe

```
JSON.ARRLEN <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la longueur du tableau à chaque chemin.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.
- Valeur nulle si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Tableau de chaînes en bloc. Chaque élément est un nom de clé dans l'objet.
- Entier, longueur du tableau.
- Si plusieurs objets sont sélectionnés, la commande renvoie la longueur du premier tableau.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.
- Erreur `WRONGTYPE` si le chemin n'existe pas.
- Valeur nulle si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], [\"a\"], [\"a\", \"b\"], [\"a\", \"b\", \"c\"]]'
```

```
(error) SYNTAXERR Failed to parse JSON string due to syntax error
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
1) (integer) 0
2) (nil)
3) (integer) 2
4) (integer) 3
5) (nil)
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k1 $[3]
1) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 $[1]
1) (nil)
127.0.0.1:6379> JSON.ARRLEN k2 $[2]
1) (integer) 2
```

JSON.ARRPOP

Supprime et renvoie l'élément à l'index du tableau. L'extraction d'un tableau vide renvoie valeur nulle.

Syntaxe

```
JSON.ARRPOP <key> [path [index]]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie
- index (facultatif) — position dans le tableau à partir de laquelle commencer à apparaître.
  - La valeur par défaut est -1 si elle n'est pas fournie, ce qui signifie le dernier élément.
  - Une valeur négative signifie une position à partir du dernier élément.
  - Les index hors limites sont arrondis à leurs limites de tableau respectives.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes groupées, représentant les valeurs affichées à chaque chemin.
- Si une valeur est un tableau vide, sa valeur de retour correspondante est nulle.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.

Si le chemin est une syntaxe restreinte :

- Chaîne en bloc, représentant la valeur JSON affichée
- Valeur nulle si le tableau est vide.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1 $[*]
1) (nil)
2) "\"a\""
3) "\"b\""
127.0.0.1:6379> JSON.GET k1
"[[],[],[\"a\"]]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1
"[\"a\",\"b\"]"
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"]]"

127.0.0.1:6379> JSON.SET k2 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k2 . 0
"[]"
127.0.0.1:6379> JSON.GET k2
"[[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRTRIM

Découpez les tableaux au niveau du chemin pour qu'il devienne un sous-réseau [début, fin], inclus dans les deux cas.

- Si le tableau est vide, ne rien faire, retourner 0.
- Si start <0, le traiter comme 0.
- Si end >= size (taille du tableau), le traiter comme size-1.
- Si start >= size ou start > end, vider le tableau et retourner 0.

Syntaxe

```
JSON.ARRINSERT <key> <path> <start> <end>
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- start (obligatoire) — index de départ inclus.
- fin (obligatoire) — index de fin, inclus.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la nouvelle longueur du tableau à chaque chemin.
- Si une valeur est un tableau vide, sa valeur de retour correspondante est nulle.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.
- Erreur `OUTOFBOUNDARIES` si un argument d'index est hors limites.

Si le chemin est une syntaxe restreinte :

- Entier, la nouvelle longueur du tableau.
- Valeur nulle si le tableau est vide.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un tableau.
- Erreur `OUTOFBOUNDARIES` si un argument d'index est hors limites.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
   127.0.0.1:6379> JSON.GET k1
   "[[],[\"a\"],[\"a\",\"b\"],[\"a\",\"b\"]]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 .children 0 1
(integer) 2
127.0.0.1:6379> JSON.GET k1 .children
"[\"John\",\"Jack\"]"
```

JSON.CLEAR

Effacez les tableaux ou les objets situés sur le chemin.

Syntaxe

```
JSON.CLEAR <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

- Entier, le nombre de conteneurs effacés.
- La suppression d'un tableau ou d'un objet vide équivaut à 0 conteneur effacé.

> **Note**
>
> Avant la version 6.2.6.R2 de Redis OSS, la suppression d'un tableau ou d'un objet vide équivaut à 1 conteneur effacé.

- L'effacement d'une valeur non-conteneur retourne 0.
- Si aucune valeur de tableau ou d'objet n'est localisée près du chemin, la commande renvoie 0.

Exemples

```
127.0.0.1:6379> JSON.SET k1 . '[[], [0], [0,1], [0,1,2], 1, true, null, "d"]'
OK
127.0.0.1:6379>  JSON.CLEAR k1  $[*]
(integer) 6
127.0.0.1:6379> JSON.CLEAR k1  $[*]
(integer) 0
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

JSON.DEBUG

Informations sur le rapport. Les sous-commandes prises en charge sont :

- MEMORY <key>[chemin] : indique l'utilisation de la mémoire en octets d'une valeur JSON. Le chemin d'accès est par défaut la racine s'il n'est pas fourni.
- <key>DEPTH [chemin] — Indique la profondeur de chemin maximale du document JSON.

> **Note**
>
> Cette sous-commande est uniquement disponible avec le moteur Redis OSS version 6.2.6.R2 ou ultérieure.

- CHAMPS <key>[chemin] : indique le nombre de champs dans le chemin du document spécifié. Le chemin d'accès est par défaut la racine s'il n'est pas fourni. Chaque valeur JSON sans conteneur compte pour un champ. Les objets et les tableaux comptent récursivement un champ pour chacune de leurs valeurs JSON contenantes. Chaque valeur de conteneur, à l'exception du conteneur racine, compte pour un champ supplémentaire.
- AIDE — affiche les messages d'aide de la commande.

Syntaxe

```
JSON.DEBUG <subcommand & arguments>
```

Dépend de la sous-commande :

MEMORY

- Si le chemin est une syntaxe améliorée :
  - renvoie un tableau d'entiers, représentant la taille de la mémoire (en octets) de la valeur JSON pour chaque chemin.
  - renvoie un tableau vide si la clé Redis OSS n'existe pas.
- Si le chemin est une syntaxe restreinte :
  - renvoie un entier, la taille de la mémoire est la valeur JSON en octets.
  - renvoie null si la clé Redis OSS n'existe pas.

DEPTH

- Renvoie un entier qui représente la profondeur de chemin maximale du document JSON.
- Renvoie null si la clé Redis OSS n'existe pas.

FIELDS

- Si le chemin est une syntaxe améliorée :
  - renvoie un tableau d'entiers, représentant le nombre de champs de valeur JSON à chaque chemin.
  - renvoie un tableau vide si la clé Redis OSS n'existe pas.
- Si le chemin est une syntaxe restreinte :
  - renvoie un entier, le nombre de champs de la valeur JSON.
  - renvoie null si la clé Redis OSS n'existe pas.

HELP — renvoie un tableau de messages d'aide.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2},
 [1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
5) (integer) 16
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
6) (integer) 0
```

```
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1
(integer) 632
127.0.0.1:6379> JSON.DEBUG MEMORY k1 .phoneNumbers
(integer) 166

127.0.0.1:6379> JSON.DEBUG FIELDS k1
(integer) 19
127.0.0.1:6379> JSON.DEBUG FIELDS k1 .address
(integer) 4

127.0.0.1:6379> JSON.DEBUG HELP
1) JSON.DEBUG MEMORY <key> [path] - report memory size (bytes) of the JSON element.
 Path defaults to root if not provided.
2) JSON.DEBUG FIELDS <key> [path] - report number of fields in the JSON element. Path
 defaults to root if not provided.
3) JSON.DEBUG HELP - print help message.
```

JSON.DEL

Supprimez les valeurs JSON dans le chemin d'une clé de document. Si le chemin est la racine, cela revient à supprimer la clé de Redis OSS.

Syntaxe

```
JSON.DEL <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

- Nombre d'éléments supprimés.
- 0 si la clé Redis OSS n'existe pas.
- 0 si le chemin JSON n'est pas valide ou n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 $.d.*
(integer) 3
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 $.e[*]
(integer) 5
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 .d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 .e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

JSON.FORGET

Un alias de JSON.DEL

JSON.GET

Renvoie le JSON sérialisé sur un ou plusieurs chemins.

Syntaxe

```
JSON.GET <key>
[INDENT indentation-string]
[NEWLINE newline-string]
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- INDENT/NEWLINE/SPACE (facultatif) — contrôle le format de la chaîne JSON renvoyée, c'est-à-dire « pretty print ». La valeur par défaut de chacune d'entre elles est une chaîne vide. Ils peuvent être annulés dans n'importe quelle combinaison. Elles peuvent être spécifiées dans n'importe quel ordre.
- NOESCAPE - facultatif, autorisé à être présent pour des raisons de compatibilité héritées et n'a aucun autre effet.
- path (facultatif) — zéro ou plusieurs chemins JSON, la valeur par défaut est la racine si aucun n'est indiqué. Les arguments de chemin doivent être placés à la fin.

Retour

Syntaxe de chemin améliorée :

Si un seul chemin est fourni :

- Renvoie une chaîne sérialisée d'un tableau de valeurs.
- Si aucune valeur n'est sélectionnée, la commande renvoie un tableau vide.

Si plusieurs chemins sont fournis :

- Renvoie un objet JSON sous forme de chaînes, dans lequel chaque chemin est une clé.

- Si la syntaxe des chemins est mixte, améliorée et restreinte, le résultat est conforme à la syntaxe améliorée.
- Si un chemin n'existe pas, sa valeur correspondante est un tableau vide.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 $.address.*
"[\"21 2nd Street\",\"New York\",\"NY\",\"10021-3100\"]"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
"[\n\t\"21 2nd Street\",\n\t\"New York\",\n\t\"NY\",\n\t\"10021-3100\"\n]"
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
"{\"$.firstName\":[\"John\"],\"$.lastName\":[\"Smith\"],\"$.age\":[27]}"
127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}}'
OK
127.0.0.1:6379> json.get k2 $..*
"[{},{\"a\":1},{\"a\":1,\"b\":2},1,1,2]"
```

Syntaxe de chemin restreinte :

```
 127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 .address
"{\"street\":\"21 2nd Street\",\"city\":\"New York\",\"state\":\"NY\",\"zipcode\":
\"10021-3100\"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
"{\n\t\"street\": \"21 2nd Street\",\n\t\"city\": \"New York\",\n\t\"state\": \"NY\",\n
\t\"zipcode\": \"10021-3100\"\n}"
```

```
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\":\"John\",\".lastName\":\"Smith\",\".age\":27}"
```

JSON.MGET

Obtenez des JSON sérialisés sur le chemin à partir de plusieurs clés de document. Renvoie null pour une clé ou un chemin JSON inexistant.

Syntaxe

```
JSON.MGET <key> [key ...] <path>
```

- clé (obligatoire) — Une ou plusieurs clés Redis OSS de type document.
- chemin (obligatoire) — un chemin JSON

Retour

- Tableau de chaînes en vrac. La taille du tableau est égale au nombre de clés dans la commande. Chaque élément du tableau est renseigné avec (a) le JSON sérialisé tel qu'il est situé par le chemin ou (b) Null si la clé n'existe pas ou si le chemin n'existe pas dans le document ou si le chemin n'est pas valide (erreur de syntaxe).
- Si l'une des clés spécifiées existe et n'est pas une clé JSON, la commande renvoie l'erreur `WRONGTYPE`.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK
127.0.0.1:6379> JSON.MGET k1 k2 k3 $.address.city
1) "[\"New York\"]"
```

```
2) "[\"Boston\"]"
3) "[\"Seattle\"]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK

127.0.0.1:6379> JSON.MGET k1 k2 k3 .address.city
1) "\"New York\""
2) "\"Seattle\""
3) "\"Seattle\""
```

JSON.NUMINCRBY

Incrémentez les valeurs numériques sur le chemin d'un nombre donné.

Syntaxe

```
JSON.NUMINCRBY <key> <path> <number>
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- numéro (obligatoire) — un chiffre

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes groupées représentant la valeur résultante pour chaque chemin.
- Si une valeur n'est pas un nombre, sa valeur de retour correspondante est nulle.

- Erreur `WRONGTYPE` si le nombre ne peut pas être analysé.
- Erreur `OVERFLOW` si le résultat est hors de la plage des doubles IEEE 64 bits.
- `NONEXISTENT` si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Chaîne en bloc représentant la valeur résultante.
- Si plusieurs valeurs sont sélectionnées, la commande renvoie le résultat de la dernière valeur mise à jour.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un nombre.
- Erreur `WRONGTYPE` si le nombre ne peut pas être analysé.
- Erreur `OVERFLOW` si le résultat est hors de la plage des doubles IEEE 64 bits.
- `NONEXISTENT` si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 10
"[11,12,13]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[11,12,13]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.a[*] 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.b[*] 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.c[*] 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"
```

```
127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 $.a.* 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.b.* 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.c.* 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.d.* 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 $.a.* 1
"[null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.b.* 1
"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":4}}"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[1] 10
"12"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,12,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .a[*] 1
(error) NONEXISTENT JSON path does not exist
```

```
127.0.0.1:6379> JSON.NUMINCRBY k1 .b[*] 1
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .c[*] 1
"3"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[*] 1
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 .a.* 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k2 .b.* 1
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .c.* 1
"3"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .d.* 1
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

JSON.NUMMULTBY

Multipliez les valeurs numériques du chemin par un nombre donné.

Syntaxe

```
JSON.NUMMULTBY <key> <path> <number>
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — un chemin JSON
- numéro (obligatoire) — un chiffre

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes groupées représentant la valeur résultante pour chaque chemin.
- Si une valeur n'est pas un nombre, sa valeur de retour correspondante est nulle.
- Erreur `WRONGTYPE` si le nombre ne peut pas être analysé.
- Erreur `OVERFLOW` si le résultat est hors de la plage des doubles IEEE 64 bits.
- `NONEXISTENT` si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Chaîne en bloc représentant la valeur résultante.
- Si plusieurs valeurs sont sélectionnées, la commande renvoie le résultat de la dernière valeur mise à jour.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas un nombre.
- Erreur `WRONGTYPE` si le nombre ne peut pas être analysé.
- Erreur `OVERFLOW` si le résultat est hors de la plage des doubles IEEE 64 bits.
- `NONEXISTENT` si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.a[*] 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.b[*] 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.c[*] 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 $.a.* 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.b.* 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.c.* 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.d.* 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 $.a.* 2
"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
"[2,null,6]"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[1] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,4,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .a[*] 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k1 .b[*] 2
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .c[*] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[*] 2
"6"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":2,\"b\":4,\"c\":6}}"
```

```
127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 .a.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":1,\"b\":\"b\",\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k3 .c.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":6}}"
```

JSON.OBJLEN

Obtenez le nombre de clés dans les valeurs des objets sur le chemin.

Syntaxe

```
JSON.OBJLEN <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la longueur de l'objet à chaque chemin.
- Si une valeur n'est pas un objet, sa valeur de retour correspondante est nulle.
- Valeur nulle si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Entier, nombre de clés dans l'objet.
- Si plusieurs objets sont sélectionnés, la commande renvoie la longueur du premier objet.
- Erreur `WRONGTYPE` si la valeur au chemin n'est pas un objet.
- Erreur `WRONGTYPE` si le chemin n'existe pas.
- Valeur nulle si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 .a
(integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 .a.*
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.OBJLEN k1 .b
(integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 .b.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .c
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .c.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .d
(integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 .d.*
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .*
(integer) 0
```

JSON.OBJKEYS

Obtenez les noms des clés dans les valeurs des objets situés sur le chemin.

Syntaxe

```
JSON.OBJKEYS <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de tableaux de chaînes en bloc. Chaque élément est un tableau de clés dans un objet correspondant.
- Si une valeur n'est pas un objet, sa valeur de retour correspondante est une valeur vide.

- Valeur nulle si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Tableau de chaînes en bloc. Chaque élément est un nom de clé dans l'objet.
- Si plusieurs objets sont sélectionnés, la commande renvoie les clés du premier objet.
- Erreur `WRONGTYPE` si la valeur au chemin n'est pas un objet.
- Erreur `WRONGTYPE` si le chemin n'existe pas.
- Valeur nulle si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 $.*
1) (empty array)
2) 1) "a"
3) 1) "a"
   2) "b"
4) 1) "a"
   2) "b"
   3) "c"
5) (empty array)
127.0.0.1:6379> JSON.OBJKEYS k1 $.d
1) 1) "a"
   2) "b"
   3) "c"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 .*
1) "a"
127.0.0.1:6379> JSON.OBJKEYS k1 .d
```

```
1) "a"
2) "b"
3) "c"
```

JSON.RESP

Renvoie la valeur JSON au chemin donné dans le protocole de sérialisation Redis OSS (RESP). Si la valeur est un conteneur, la réponse est un tableau RESP ou un tableau imbriqué.

- La valeur nulle de JSON est mappée à Null Bulk String de RESP.
- Les valeurs booléennes JSON sont mappées aux chaînes simples RESP respectives.
- Les nombres entiers sont mappés aux entiers de RESP.
- Les nombres à virgule flottante double IEEE de 64 bits sont mappés aux chaînes en bloc de RESP.
- Les chaînes JSON sont mappées aux chaînes en vrac RESP.
- Les tableaux JSON sont représentés sous forme de tableaux RESP, où le premier élément est la simple chaîne [, suivie des éléments du tableau.
- Les objets JSON sont représentés sous forme de tableaux RESP, où le premier élément est la chaîne simple {, suivie de paires clé-valeur, chacune étant une chaîne en vrac RESP.

Syntaxe

```
JSON.RESP <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de tableaux. Chaque élément du tableau représente la forme RESP de la valeur au niveau d'un chemin.
- Tableau vide si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Tableau, représentant la forme RESP de la valeur sur le chemin.
- Valeur nulle si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 $.address
1) 1) {
   2) 1) "street"
      2) "21 2nd Street"
   3) 1) "city"
      2) "New York"
   4) 1) "state"
      2) "NY"
   5) 1) "zipcode"
      2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.address.*
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
1) 1) [
   2) 1) {
      2) 1) "type"
         2) "home"
      3) 1) "number"
         2) "555 555-1234"
   3) 1) {
      2) 1) "type"
         2) "office"
```

```
      3) 1) "number"
         2) "555 555-4567"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
1) 1) {
   2) 1) "type"
      2) "home"
   3) 1) "number"
      2) "212 555-1234"
2) 1) {
   2) 1) "type"
      2) "office"
   3) 1) "number"
      2) "555 555-4567"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 .address
1) {
2) 1) "street"
   2) "21 2nd Street"
3) 1) "city"
   2) "New York"
4) 1) "state"
   2) "NY"
5) 1) "zipcode"
   2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1
 1) {
 2) 1) "firstName"
    2) "John"
 3) 1) "lastName"
    2) "Smith"
```

```
 4) 1) "age"
    2) (integer) 27
 5) 1) "weight"
    2) "135.25"
 6) 1) "isAlive"
    2) true
 7) 1) "address"
    2) 1) {
       2) 1) "street"
          2) "21 2nd Street"
       3) 1) "city"
          2) "New York"
       4) 1) "state"
          2) "NY"
       5) 1) "zipcode"
          2) "10021-3100"
 8) 1) "phoneNumbers"
    2) 1) [
       2) 1) {
          2) 1) "type"
             2) "home"
          3) 1) "number"
             2) "212 555-1234"
       3) 1) {
          2) 1) "type"
             2) "office"
          3) 1) "number"
             2) "555 555-4567"
 9) 1) "children"
    2) 1) [
10) 1) "spouse"
    2) (nil)
```

JSON.SET

Définissez des valeurs JSON sur le chemin.

Si le chemin fait appel à un membre d'objet :

- Si l'élément parent n'existe pas, la commande renvoie une erreur NON EXISTANTE.
- Si l'élément parent existe mais n'est pas un objet, la commande renvoie ERROR.
- Si l'élément parent existe et est un objet :

- Si l'élément n'existe pas, un nouvel élément sera ajouté à l'objet parent si et seulement si l'objet parent est le dernier enfant dans le chemin. Sinon, la commande renverra une erreur INEXISTANTE.
- Si le membre existe, sa valeur sera remplacée par la valeur JSON.

Si le chemin fait appel à un index de tableau :

- Si l'élément parent n'existe pas, la commande renvoie une erreur INEXISTANTE.
- Si l'élément parent existe mais n'est pas un tableau, la commande renvoie ERROR.
- Si l'élément parent existe mais que l'index est hors limites, la commande renvoie l'erreur OUTOFBOUNDARIES.
- Si l'élément parent existe et que l'index est valide, l'élément sera remplacé par la nouvelle valeur JSON.

Si le chemin fait appel à un objet ou à un tableau, la valeur (objet ou tableau) sera remplacée par la nouvelle valeur JSON.

Syntaxe

```
JSON.SET <key> <path> <json> [NX | XX]
```

[NX | XX] Où vous pouvez avoir 0 ou 1 des identifiants [NX | XX]

- clé (obligatoire) — clé Redis OSS de type document JSON
- chemin (obligatoire) — chemin JSON. Pour une nouvelle clé Redis OSS, le chemin JSON doit être la racine «. ».
- NX (facultatif) — Si le chemin est la racine, définissez la valeur uniquement si la clé Redis OSS n'existe pas, c'est-à-dire insérez un nouveau document. Si le chemin n'est pas la racine, définissez la valeur uniquement si le chemin n'existe pas, c'est-à-dire insérez une valeur dans le document.
- XX (facultatif) — Si le chemin est la racine, définissez la valeur uniquement si la clé Redis OSS existe, c'est-à-dire remplacez le document existant. Si le chemin n'est pas la racine, définissez la valeur uniquement si le chemin existe, c'est-à-dire mettez à jour la valeur existante.

Retour

- Simple String « OK » en cas de succès.

- Valeur nulle si la condition NX ou XX n'est pas remplie.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.SET k1 $.a.* '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"

127.0.0.1:6379> JSON.SET k2 . '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k2 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":[0,0,0,0,0]}"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"c":{"a":1, "b":2}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k1 .c.a '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.SET k1 .e[-1] '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,0]}"
127.0.0.1:6379> JSON.SET k1 .e[5] '0'
(error) OUTOFBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Ajoutez une chaîne aux chaînes JSON au niveau du chemin.

Syntaxe

```
JSON.STRAPPEND <key> [path] <json_string>
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie
- json_string (obligatoire) — Représentation JSON d'une chaîne. Notez qu'une chaîne JSON doit être entre guillemets, c'est-à-dire « foo"'.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la nouvelle longueur de la chaîne à chaque chemin.
- Si une valeur au niveau du chemin n'est pas une chaîne, sa valeur de retour correspondante est nulle.
- `SYNTAXERR` erreur si l'argument json d'entrée n'est pas une chaîne JSON valide.
- `NONEXISTENT` erreur si le chemin n'existe pas.

Si le chemin est une syntaxe restreinte :

- Entier, la nouvelle longueur de la chaîne.
- Si plusieurs valeurs de chaîne sont sélectionnées, la commande renvoie la nouvelle longueur de la dernière chaîne mise à jour.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas une chaîne.
- Erreur `WRONGTYPE` si l'argument json en entrée n'est pas une chaîne JSON valide.
- Erreur `NONEXISTENT` si le chemin n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a '"a"'
1) (integer) 2
```

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* '"a"'
1) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* '"a"'
1) (integer) 2
2) (nil)
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* '"a"'
1) (integer) 2
2) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b '"a"'
1) (integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* '"a"'
1) (nil)
2) (integer) 2
3) (nil)
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b '"a"'
(integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* '"a"'
(integer) 2
```

JSON.STRLEN

Obtenez les longueurs des valeurs de chaîne JSON sur le chemin.

Syntaxe

```
JSON.STRLEN <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers, représentant la longueur de la valeur de chaîne à chaque chemin.
- Si une valeur n'est pas une chaîne, sa valeur de retour correspondante est nulle.
- Valeur nulle si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Entier, la longueur de la chaîne.
- Si plusieurs valeurs de chaîne sont sélectionnées, la commande renvoie la longueur de la première chaîne.
- Erreur `WRONGTYPE` si la valeur au niveau du chemin n'est pas une chaîne.
- Erreur `NONEXISTENT` si le chemin n'existe pas.
- Valeur nulle si la clé du document n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
1) (integer) 1
2) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
1) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
1) (nil)
```

```
2) (integer) 1
3) (nil)
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 .a.a
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .a.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.b
(integer) 2
127.0.0.1:6379> JSON.STRLEN k1 .d.*
(integer) 1
```

JSON.TOGGLE

Basculez les valeurs booléennes entre vrai et faux sur le chemin.

Syntaxe

```
JSON.TOGGLE <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers (0 - faux, 1 - vrai) représentant la valeur booléenne résultante pour chaque chemin.
- Si une valeur n'est pas une valeur booléenne, la valeur de retour correspondante est nulle.
- `NONEXISTENT` si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Chaîne (« true » /"false ») représentant la valeur booléenne résultante.
- `NONEXISTENT` si la clé du document n'existe pas.
- `WRONGTYPE`erreur si la valeur du chemin n'est pas une valeur booléenne.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '{"a":true, "b":false, "c":1, "d":null, "e":"foo", "f":
[], "g":{}}'
OK
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 0
2) (integer) 1
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 1
2) (integer) 0
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 . true
OK
127.0.0.1:6379> JSON.TOGGLE k1
"false"
127.0.0.1:6379> JSON.TOGGLE k1
"true"

127.0.0.1:6379> JSON.SET k2 . '{"isAvailable": false}'
```

```
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

JSON.TYPE

Type de rapport des valeurs sur le chemin donné.

Syntaxe

```
JSON.TYPE <key> [path]
```

- clé (obligatoire) — clé Redis OSS de type document JSON
- path (facultatif) — un chemin JSON. La valeur par défaut est la racine si elle n'est pas fournie

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes représentant le type de valeur de chaque chemin. Le type est l'un de {"null", "boolean", "string", "number", "integer", "object" et "array"}.
- Si un chemin n'existe pas, sa valeur de retour correspondante est nulle.
- Tableau vide si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Chaîne, type de la valeur
- Valeur nulle si la clé du document n'existe pas.
- Valeur nulle si le chemin JSON n'est pas valide ou n'existe pas.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
```

```
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.TYPE k1
object
127.0.0.1:6379> JSON.TYPE k1 .children
array
127.0.0.1:6379> JSON.TYPE k1 .firstName
string
127.0.0.1:6379> JSON.TYPE k1 .age
integer
127.0.0.1:6379> JSON.TYPE k1 .weight
number
127.0.0.1:6379> JSON.TYPE k1 .isAlive
boolean
127.0.0.1:6379> JSON.TYPE k1 .spouse
null
```

Marquer vos ressources MemoryDB

Pour vous aider à gérer vos clusters et autres ressources MemoryDB, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Les balises vous permettent de classer vos AWS ressources de différentes manières, par exemple par objectif, propriétaire ou environnement. Cette approche est utile lorsque vous avez de nombreuses ressources de même

type. Elle vous permet d'identifier rapidement une ressource spécifique en fonction des balises que vous lui avez attribuées. Cette rubrique décrit les balises et vous montre comment les créer.

> **Warning**
>
> Nous vous recommandons de ne pas inclure de données sensibles dans vos balises.

Principes de base des étiquettes

Une étiquette est une étiquette que vous attribuez à une AWS ressource. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Les balises vous permettent de classer vos AWS ressources de différentes manières, par exemple par objectif ou par propriétaire. Par exemple, vous pouvez définir un ensemble de balises pour les clusters MemoryDB de votre compte afin de suivre le propriétaire et le groupe d'utilisateurs de chaque cluster.

Nous vous recommandons de concevoir un ensemble de clés d'étiquette répondant à vos besoins pour chaque type de ressource. L'utilisation d'un ensemble de clés de balise cohérent facilite la gestion de vos ressources. Vous pouvez rechercher et filtrer les ressources en fonction des balises que vous ajoutez. Pour plus d'informations sur la mise en œuvre d'une stratégie efficace de balisage des ressources, consultez Le livre blanc AWS sur les bonnes pratiques en matière d'identification.

Les balises n'ont aucune signification sémantique pour MemoryDB et sont interprétées strictement comme une chaîne de caractères. De plus, les étiquettes ne sont pas automatiquement affectées à vos ressources. Vous pouvez modifier les clés et valeurs de balise, et vous pouvez retirer des balises d'une ressource à tout moment. Vous pouvez définir la valeur d'une balise à `null`. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur. Si vous supprimez une ressource, ses balises sont également supprimées.

Vous pouvez travailler avec des balises à l'aide de l' AWS Management Console API, du AWS CLI, et de l'API MemoryDB.

Si vous utilisez IAM, vous pouvez contrôler quels utilisateurs de votre AWS compte sont autorisés à créer, modifier ou supprimer des tags. Pour plus d'informations, consultez Autorisations de niveau ressource.

Ressources que vous pouvez étiqueter

Vous pouvez baliser la plupart des ressources MemoryDB qui existent déjà dans votre compte. Le tableau ci-dessous répertorie les ressources qui prennent en charge le balisage. Si vous utilisez

le AWS Management Console, vous pouvez appliquer des balises aux ressources à l'aide de l'éditeur de balises. Certains écrans de ressources vous permettent de spécifier des balises pour une ressource lorsque vous la créez ; par exemple, une balise avec une clé de Nom et une valeur que vous spécifiez. Dans la plupart des cas, la console applique les balises immédiatement après la création de la ressource (plutôt qu'au cours de la création de ressources). La console peut organiser les ressources en fonction de la balise Name, mais cette balise n'a aucune signification sémantique pour le service MemoryDB.

En outre, certaines actions de création de ressources vous permettent de spécifier des balises pour une ressource lors de la création de cette dernière. Si les balises ne peuvent pas être appliquées au cours de la création de ressources, nous restaurons le processus de création de ressources. Cela permet de s'assurer que les ressources sont créées avec des balises ou qu'elles ne sont pas créées du tout, et qu'aucune ressource ne demeurent sans balise à tout moment. En attribuant des balises aux ressources au moment de la création, vous pouvez supprimer la nécessité d'exécuter des scriptes de balisage personnalisés après la création de ressources.

Si vous utilisez l'API Amazon MemoryDB, la AWS CLI ou un AWS SDK, vous pouvez utiliser le `Tags` paramètre de l'action d'API MemoryDB correspondante pour appliquer des balises. Il s'agit des options suivantes :

- `CreateCluster`
- `CopySnapshot`
- `CreateParameterGroup`
- `CreateSubnetGroup`
- `CreateSnapshot`
- `CreateACL`
- `CreateUser`

Le tableau suivant décrit les ressources MemoryDB qui peuvent être balisées et les ressources qui peuvent être étiquetées lors de la création à l'aide de l'API MemoryDB, de la AWS CLI ou d'un SDK. AWS

Support de balisage pour les ressources MemoryDB

Prend en charge les balises	Prend en charge le balisage au moment de la création
Oui	Oui
Oui	Oui
Oui	Oui
Oui	Oui
Oui	Oui
Oui	Oui

Vous pouvez appliquer des autorisations au niveau des ressources basées sur des balises dans vos politiques IAM aux actions de l'API MemoryDB qui prennent en charge le balisage lors de la création afin de mettre en œuvre un contrôle granulaire sur les utilisateurs et les groupes autorisés à étiqueter les ressources lors de la création. Vos ressources sont correctement sécurisées depuis la création. Les balises sont appliquées immédiatement à vos ressources. Les autorisations de niveau ressource basées sur des balises sont donc effectives immédiatement. Vos ressources peuvent être suivies et signalées avec plus de précision. Vous pouvez appliquer l’utilisation du balisage sur les nouvelles ressources et contrôler que les clés et valeurs de balise sont définies sur vos ressources.

Pour plus d’informations, consultez Exemple : étiquetage de vos ressources.

Pour plus d'informations sur l'étiquetage de vos ressources pour la facturation, veuillez consulter Surveillance des coûts avec des balises de répartition des coûts.

Marquage des clusters et des instantanés

Les règles suivantes s'appliquent à l'étiquetage dans le cadre d'opérations de requête :

- CreateCluster :
  - Si `--cluster-name` est fourni :

    Si des balises sont incluses dans la demande, le cluster sera balisé.
  - Si `--snapshot-name` est fourni :

    Si des balises sont incluses dans la demande, le cluster ne sera étiqueté qu'avec ces balises. Si aucune balise n'est incluse dans la demande, les balises de capture d'écran seront ajoutées au cluster.
- CreateSnapshot :
  - Si `--cluster-name` est fourni :

    Si des balises sont incluses dans la requête, seules les balises de requête seront ajoutées à l'instantané. Si aucune balise n'est incluse dans la demande, les balises de cluster seront ajoutées à l'instantané.
  - Pour les instantanés automatiques :

    Les balises se propageront à partir des balises du cluster.
- CopySnapshot :

  Si des balises sont incluses dans la requête, seules les balises de requête seront ajoutées à l'instantané. Si aucune balise n'est incluse dans la requête, les balises d'instantané source sont ajoutées à l'instantané copié.
- TagResourceet UntagResource:

  Des balises seront ajoutées/supprimées de la ressource.

Restrictions liées aux étiquettes

Les restrictions de base suivantes s’appliquent aux balises :

- Nombre maximal de balises par ressource : 50
- Pour chaque ressource, chaque clé de balise doit être unique, et chaque clé de balise peut avoir une seule valeur.
- Longueur de clé maximale : 128 caractères Unicode en UTF-8.
- Longueur de valeur maximale : 256 caractères Unicode en UTF-8.

- Bien que MemoryDB autorise n'importe quel caractère dans ses balises, d'autres services peuvent être restrictifs. Les caractères autorisés pour les services sont les lettres, les chiffres et les espaces représentables en UTF-8, ainsi que les caractères suivants : + - = . _ : / @
- Les clés et valeurs d'étiquette sont sensibles à la casse.
- Le `aws:` préfixe est réservé à l' AWS usage. Lorsque la balise possède une clé de balise avec ce préfixe, vous ne pouvez pas modifier ou supprimer sa clé ou sa valeur. Les balises avec le préfixe `aws:` ne sont pas comptabilisées comme vos balises pour la limite de ressources.

Vous ne pouvez pas mettre fin à une ressource, ou l'arrêter ou la supprimer uniquement en fonction de ses balises ; vous devez spécifier l'identificateur de ressource. Par exemple, pour supprimer des instantanés (snapshot) que vous avez balisés avec une clé de balise appelée `DeleteMe`, vous devez utiliser l'action `DeleteSnapshot` avec les identificateurs de ressource des instantanés, tels que `snap-1234567890abcdef0`.

Pour plus d'informations sur les ressources MemoryDB que vous pouvez baliser, consultez. Ressources que vous pouvez étiqueter

Exemple : étiquetage de vos ressources

- Ajouter des balises à un cluster.

```
aws memorydb tag-resource \
--resource-arn arn:aws:memorydb:us-east-1:111111222233:cluster/my-cluster \
--tags Key="project",Value="XYZ" Key="memorydb",Value="Service"
```

- Création d'un cluster à l'aide de balises

```
aws memorydb create-cluster \
--cluster-name testing-tags \
--description cluster-test \
--subnet-group-name test \
--node-type db.r6g.large \
--acl-name open-access \
--tags Key="project",Value="XYZ" Key="memorydb",Value="Service"
```

- Création d'un instantané avec des balises.

  Dans ce cas, si vous ajoutez des balises sur demande, même si le cluster contient des balises, l'instantané ne recevra que les balises de demande.

```
aws memorydb create-snapshot \
--cluster-name testing-tags \
--snapshot-name bkp-testing-tags-mycluster \
--tags Key="work",Value="foo"
```

Surveillance des coûts avec des balises de répartition des coûts

Lorsque vous ajoutez des balises de répartition des coûts à vos ressources dans MemoryDB, vous pouvez suivre les coûts en regroupant les dépenses sur vos factures par valeur d'étiquette de ressource.

Une balise de répartition des coûts MemoryDB est une paire clé-valeur que vous définissez et associez à une ressource MemoryDB. Les clés et les valeurs sont sensibles à la casse. Vous pouvez utiliser une clé de balise pour définir une catégorie, et la valeur de balise peut être un élément de cette catégorie. Par exemple, vous pouvez définir une clé de balise appelée `CostCenter` et une valeur de balise appelée `10010`, en indiquant que la ressource est assignée au centre de coûts 10010. Vous pouvez également utiliser des balises pour désigner des ressources destinées aux tests ou à la production en utilisant une clé telle que `Environment` et des valeurs telles que `test` ou `production`. Pour faciliter le suivi des coûts associés à vos ressources, nous vous recommandons d'utiliser un ensemble de clés de balise cohérent.

Utilisez des balises de répartition des coûts 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. Ensuite, pour voir le coût de vos ressources combinées, organisez vos informations de facturation en fonction des ressources possédant les mêmes valeurs de clé de balise. 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.

Vous pouvez également combiner des balises pour suivre les coûts plus détaillés. Par exemple, pour suivre vos coûts de service par région, vous pouvez utiliser les clés de balise `Service` et `Region`. Sur une seule ressource, vous pouvez avoir les valeurs `MemoryDB` et `Asia Pacific (Singapore)`, et sur une autre ressource, les valeurs `MemoryDB` et `Europe (Frankfurt)`. Vous pouvez ensuite voir vos coûts totaux de MemoryDB répartis par région. Pour de plus amples informations, veuillez consulter Utilisation des balises d’allocation des coûts dans le Guide de l'utilisateur AWS Billing .

Vous pouvez ajouter des balises de répartition des coûts MemoryDB aux clusters MemoryDB. Lorsque vous ajoutez, affichez, modifiez, copiez ou supprimez une balise, l'opération est appliquée uniquement au cluster spécifié.

Caractéristiques des balises de répartition des coûts MemoryDB

- Les balises de répartition des coûts sont appliquées aux ressources MemoryDB spécifiées dans les opérations de la CLI et de l'API sous forme d'ARN. Le type de ressource sera un « cluster ».

  Format de l'ARN : `arn:aws:memorydb:<region>:<customer-id>:<resource-type>/<resource-name>`

  Exemple d'ARN : `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`
- La clé de balise correspond au nom obligatoire de la balise. La valeur de la chaîne de caractères de la clé peut comporter de 1 à 128 caractères Unicode et ne peut pas être précédée de `aws:`. La chaîne peut uniquement contenir l'ensemble de lettres, de chiffres et d'espaces, de traits de soulignement (_), de points (.), de deux-points (:), de barres obliques inverses (\ ), de signes égal (=), de signes plus (+), de tirets (-) ou d'arobases (@).
- La valeur de balise est la valeur facultative de la balise. La valeur de la chaîne de caractères de la chaîne peut comporter de 1 à 256 caractères Unicode, et ne peut pas être précédée de `aws:`. La chaîne peut uniquement contenir l'ensemble de lettres, de chiffres et d'espaces, de traits de soulignement (_), de points (.), de deux-points (:), de barres obliques inverses (\ ), de signes égal (=), de signes plus (+), de tirets (-) ou d'arobases (@).
- Une ressource MemoryDB peut comporter un maximum de 50 balises.
- Les valeurs comprises dans un ensemble de balises, ne doivent pas nécessairement être uniques. Par exemple, vous pouvez avoir une balise définie où les clés `Service` et `Application` ont, toutes deux, la valeur `MemoryDB`.

AWS n'applique aucune signification sémantique à vos balises. Les balises sont interprétées strictement comme des chaînes de caractères. AWS ne définit pas automatiquement de balises sur aucune ressource MemoryDB.

Gérez vos étiquettes de répartition des coûts à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour ajouter, modifier ou supprimer des balises de répartition des coûts.

Exemple d'ARN : `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Rubriques

- Lister les tags à l'aide du AWS CLI
- Ajout de balises à l'aide du AWS CLI
- Modification des balises à l'aide du AWS CLI
- Suppression de balises à l'aide du AWS CLI

Lister les tags à l'aide du AWS CLI

Vous pouvez utiliser les balises AWS CLI to list sur une ressource MemoryDB existante en utilisant l'opération list-tags.

Le code suivant utilise le AWS CLI pour répertorier les balises du cluster MemoryDB `my-cluster` dans la région us-east-1.

Pour Linux, macOS ou Unix :

```
aws memorydb list-tags \
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

Pour Windows :

```
aws memorydb list-tags ^
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

Le résultat de cette opération se présentera de la façon suivante, une liste de toutes les balises sur la ressource.

```
{
   "TagList": [
      {
         "Value": "10110",
         "Key": "CostCenter"
      },
      {
         "Value": "EC2",
         "Key": "Service"
      }
   ]
}
```

S'il n'y a pas de balises sur la ressource, la sortie sera vide TagList.

```
{
   "TagList": []
}
```

Pour plus d'informations, consultez les balises de AWS CLI liste for MemoryDB.

Ajout de balises à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour ajouter des balises à une ressource MemoryDB existante à l'aide de l'opération tag-resource CLI. Si la clé de balise n'existe pas sur la ressource, la clé et la valeur sont ajoutées à la ressource. Si la clé existe déjà sur la ressource, la valeur associée à cette clé est mise à jour en la nouvelle valeur.

Le code suivant utilise le AWS CLI pour ajouter les clés `Service` et `Region` avec les valeurs`memorydb`, `us-east-1` respectivement au cluster de la région `my-cluster` us-east-1.

Pour Linux, macOS ou Unix :

```
aws memorydb tag-resource \
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster \
 --tags Key=Service,Value=memorydb \
        Key=Region,Value=us-east-1
```

Pour Windows :

```
aws memorydb tag-resource ^
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster ^
 --tags Key=Service,Value=memorydb ^
        Key=Region,Value=us-east-1
```

Le résultat de cette commande se présentera de la façon suivante, une liste de toutes les balises sur la ressource à la suite de l'opération.

```
{
   "TagList": [
      {
         "Value": "memorydb",
         "Key": "Service"
      },
```

```
      {
         "Value": "us-east-1",
         "Key": "Region"
      }
   ]
}
```

Pour plus d'informations, consultez le AWS CLI for MemoryDBtag-resource.

Vous pouvez également utiliser le AWS CLI pour ajouter des balises à un cluster lorsque vous créez un nouveau cluster à l'aide de l'opération create-cluster.

Modification des balises à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour modifier les balises d'un cluster MemoryDB.

Pour modifier des balises :

- Utilisez tag-resource pour ajouter une nouvelle balise et une nouvelle valeur ou pour modifier la valeur associée à une balise existante.
- Utilisez untag-resource pour supprimer les balises spécifiées de la ressource.

Le résultat de l'une ou l'autre de ces opérations sera une liste de toutes les balises et de leurs valeurs sur le cluster spécifié.

Suppression de balises à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour supprimer des balises d'un cluster MemoryDB existant en utilisant l'opération untag-resource.

Le code suivant utilise le AWS CLI pour supprimer les balises à l'aide des clés `Service` et `Region` du cluster `my-cluster` dans la région us-east-1.

Pour Linux, macOS ou Unix :

```
aws memorydb untag-resource \
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster \
 --tag-keys Region Service
```

Pour Windows :

```
aws memorydb untag-resource ^
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster ^
 --tag-keys Region Service
```

Le résultat de cette commande se présentera de la façon suivante, une liste de toutes les balises sur la ressource à la suite de l'opération.

```
{
   "TagList": []
}
```

Pour plus d'informations, consultez le AWS CLI for MemoryDB untag-resource.

Gestion de vos balises de répartition des coûts à l'aide de l'API MemoryDB

Vous pouvez utiliser l'API MemoryDB pour ajouter, modifier ou supprimer des balises de répartition des coûts.

Les balises de répartition des coûts sont appliquées à MemoryDB pour les clusters. Le cluster à étiqueter est spécifié à l'aide d'un ARN (Amazon Resource Name).

Exemple d'ARN : `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Rubriques

- Lister les balises à l'aide de l'API MemoryDB
- Ajout de balises à l'aide de l'API MemoryDB
- Modification des balises à l'aide de l'API MemoryDB
- Suppression de balises à l'aide de l'API MemoryDB

Lister les balises à l'aide de l'API MemoryDB

Vous pouvez utiliser l'API MemoryDB pour répertorier les balises d'une ressource existante à l'aide de l'ListTagsopération.

Le code suivant utilise l'API MemoryDB pour répertorier les balises de la ressource `my-cluster` dans la région us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=ListTags
```

```
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2021-01-01
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Ajout de balises à l'aide de l'API MemoryDB

Vous pouvez utiliser l'API MemoryDB pour ajouter des balises à un cluster MemoryDB existant en utilisant l'opération. TagResource Si la clé de balise n'existe pas sur la ressource, la clé et la valeur sont ajoutées à la ressource. Si la clé existe déjà sur la ressource, la valeur associée à cette clé est mise à jour en la nouvelle valeur.

Le code suivant utilise l'API MemoryDB pour ajouter les clés `Service` et les valeurs `memorydb` `us-east-1` respectivement `Region` à la ressource `my-cluster` dans la région us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=TagResource
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=memorydb
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-east-1
   &Version=2021-01-01
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez TagResource.

Modification des balises à l'aide de l'API MemoryDB

Vous pouvez utiliser l'API MemoryDB pour modifier les balises d'un cluster MemoryDB.

Pour modifier la valeur d'une balise :

- Utilisez TagResourcel'opération pour ajouter une nouvelle balise et une nouvelle valeur ou pour modifier la valeur d'une balise existante.
- UntagResourceÀ utiliser pour supprimer des balises de la ressource.

Le résultat de l'une ou l'autre de ces opérations sera une liste de toutes les balises et leurs valeurs sur la ressource spécifiée.

Suppression de balises à l'aide de l'API MemoryDB

Vous pouvez utiliser l'API MemoryDB pour supprimer des balises d'un cluster MemoryDB existant en utilisant l'opération. UntagResource

Le code suivant utilise l'API MemoryDB pour supprimer les balises contenant les clés `Service` et `Region` du cluster `my-cluster` dans la région us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=UntagResource
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &TagKeys.member.1=Service
   &TagKeys.member.2=Region
   &Version=2021-01-01
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Gestion de la maintenance

Chaque cluster a un créneau de maintenance hebdomadaire au cours duquel toutes les modifications systèmes seront appliquées. Si vous ne spécifiez pas de fenêtre de maintenance préférée lorsque vous créez ou modifiez un cluster, MemoryDB attribue une fenêtre de maintenance de 60 minutes dans le créneau de maintenance de votre région, un jour de la semaine choisi au hasard.

Ce créneau de maintenance de 60 minutes est choisi de manière aléatoire sur un bloc horaire de 8 heures par région. Le tableau suivant répertorie pour les différentes régions les blocs de temps à partir desquels les créneaux de maintenance par défaut sont alloués. Vous pouvez choisir un créneau de maintenance préféré en dehors du créneau de maintenance de votre région.

Code région	Nom de la région	Fenêtre de maintenance régionale
`ap-northeast-1`	Région Asie-Pacifique (Tokyo)	13:00–21:00 UTC
`ap-northeast-2`	Région Asia Pacific (Seoul)	12:00–20:00 UTC

Code région	Nom de la région	Fenêtre de maintenance régionale
ap-south-1	Région Asie-Pacifique (Mumbai)	17:30–01:30 UTC
ap-southeast-1	Région Asie-Pacifique (Singapour)	14:00–22:00 UTC
ap-east-1	Région Asie-Pacifique (Hong Kong)	13:00–21:00 UTC
ap-southeast-2	Région Asie-Pacifique (Sydney)	12:00–20:00 UTC
cn-north-1	Région Chine (Beijing)	14:00–22:00 UTC
cn-northwest-1	Région Chine (Ningxia)	14:00–22:00 UTC
eu-west-3	Région Europe (Paris)	23:59–07:29 UTC
eu-central-1	Région Europe (Francfort)	23:00–07:00 UTC
eu-west-1	Région Europe (Irlande)	22:00–06:00 UTC
eu-west-2	Région Europe (Londres)	23:00–07:00 UTC
sa-east-1	Région Amérique du Sud (São Paulo)	01:00–09:00 UTC
ca-central-1	Région Canada (Centre)	03:00–11:00 UTC
us-east-1	Région USA Est (Virginie du Nord)	03:00–11:00 UTC
us-east-1	Région US East (Ohio)	04:00–12:00 UTC
us-west-1	Région US West (N. California)	06:00–14:00 UTC
us-west-2	Région USA Ouest (Oregon)	06:00–14:00 UTC

Modification du créneau de maintenance de votre cluster

Le créneau de maintenance doit intervenir au moment où l'utilisation est la plus faible et peut donc nécessiter d'être modifié de temps en temps. Vous pouvez modifier votre cluster en spécifiant une plage de temps de 24 heures au cours de laquelle toutes les opérations de maintenance demandées

doivent avoir lieu. Toute modification de cluster en suspens ou différé demandée doit avoir lieu au cours de cette période.

En savoir plus

Pour plus d'informations sur votre créneau de maintenance et le remplacement des nœuds, veuillez consulter :

- Remplacement de nœuds — Gestion du remplacement des nœuds
- Modification d'un cluster MemoryDB — Modification du créneau de maintenance d'un cluster

Bonnes pratiques

Vous trouverez ci-dessous les meilleures pratiques recommandées pour MemoryDB. La mise en œuvre de ces bonnes pratiques améliore les performances et la fiabilité de votre cluster.

Rubriques

- Commandes Redis OSS restreintes
- Résilience dans MemoryDB
- Bonnes pratiques : Pub/Sub et multiplexage d'E/S améliorées
- Bonnes pratiques : redimensionnement des clusters en ligne

Commandes Redis OSS restreintes

Pour offrir une expérience de service géré, MemoryDB restreint l'accès à certaines commandes qui nécessitent des privilèges avancés. Les commandes suivantes ne sont pas disponibles :

- `acl deluser`
- `acl load`
- `acl save`
- `acl setuser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster delslot`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `module`
- `psync`
- `replicaof`
- `save`
- `shutdown`
- `slaveof`
- `sync`

Résilience dans MemoryDB

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 mondiale.

Outre l'infrastructure AWS globale, MemoryDB propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience des données et de capture instantanée.

Rubriques

- Atténuation des défaillances

Atténuation des défaillances

Lorsque vous planifiez votre implémentation de MemoryDB, vous devez planifier de manière à ce que les défaillances aient un impact minimal sur votre application et vos données. Les rubriques dans cette section présentent les approches que vous pouvez entreprendre pour éviter d'éventuelles défaillances de vos applications et données.

Atténuation des défaillances : clusters MemoryDB

Un cluster MemoryDB est composé d'un seul nœud principal sur lequel votre application peut à la fois lire et écrire, et de 0 à 5 nœuds de réplication en lecture seule. Toutefois, nous vous recommandons vivement d'utiliser au moins une réplique pour une haute disponibilité. Chaque fois que des données sont écrites sur le nœud principal, elles sont conservées dans le journal des transactions et mises à jour de manière asynchrone sur les nœuds répliques.

Cas d'échec d'un réplica en lecture

1. MemoryDB détecte la réplique défaillante.
2. MemoryDB met le nœud défaillant hors ligne.
3. MemoryDB lance et approvisionne un nœud de remplacement dans le même AZ.

4. Le nouveau nœud se synchronise avec le journal des transactions.

Pendant ce temps, votre application peut continuer à lire et à écrire en utilisant les autres nœuds.

MemoryDB Multi-AZ

Si Multi-AZ est activé sur vos clusters MemoryDB, un primaire défaillant sera détecté et remplacé automatiquement.

1. MemoryDB détecte la défaillance du nœud principal.
2. MemoryDB bascule vers une réplique après s'être assurée qu'elle est cohérente avec le primaire défaillant.
3. MemoryDB lance une réplique dans l'AZ du serveur principal défaillant.
4. Le nouveau nœud se synchronise avec le journal des transactions.

Le basculement vers un nœud de réplica est généralement plus rapide que la création et la mise en service d'un nouveau nœud principal. Cela signifie que votre application peut reprendre l'écriture sur votre nœud principal plus rapidement.

Pour plus d'informations, voir Minimiser les temps d'arrêt dans MemoryDB avec Multi-AZ.

Bonnes pratiques : Pub/Sub et multiplexage d'E/S améliorées

Lorsque vous utilisez Redis OSS version 7 ou ultérieure, nous vous recommandons d'utiliser un Pub/Sub fragmenté. Vous améliorez également le débit et la latence grâce au multiplexage d'E/S amélioré, qui est automatiquement disponible lors de l'utilisation de Redis OSS version 7 ou ultérieure et ne nécessite aucune modification du client. Il est idéal pour les charges de travail pub/sub, qui sont souvent limitées en termes de débit en raison de différentes connexions client.

Bonnes pratiques : redimensionnement des clusters en ligne

Le repartitionnement implique l'ajout de partitions ou de nœuds à votre cluster, ou leur suppression, et la redistribution des espaces clés. En conséquence, plusieurs aspects peuvent avoir un impact sur l'opération de repartitionnement, tels que la charge sur le cluster, l'utilisation de la mémoire et la taille globale des données. Pour bénéficier de la meilleure expérience possible, il est recommandé de suivre les bonnes pratiques générales relatives au cluster en vue d'une distribution uniforme des modèles de charge de travail. En outre, il est recommandé de respecter les étapes suivantes.

Avant de lancer le repartitionnement, procédez comme suit :

- Testez votre application – Testez le comportement de votre application lors du repartitionnement dans un environnement intermédiaire si possible.
- Obtenez une notification anticipée pour les problèmes de mise à l'échelle – Le repartitionnement est une opération gourmande en calculs. C'est pourquoi nous recommandons de maintenir l'utilisation du processeur en dessous de 80 % sur les instances multicœurs et à moins de 50 % sur les instances monocœurs lors du repartage. Surveillez les métriques de MemoryDB et initiez le repartage avant que votre application ne commence à détecter des problèmes de dimensionnement. Les métriques qu'il est utile de suivre sont `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` et `BytesUsedForMemoryDB`.
- Assurez-vous qu'une mémoire suffisante est disponible avant de procéder à une diminution d'échelle – Si vous procédez à une diminution d'échelle, assurez-vous que cette mémoire disponible sur les partitions à conserver est au moins égale à une fois et demi la mémoire utilisée sur les partitions que vous prévoyez de supprimer.
- Initiez le repartitionnement pendant les heures creuses – Cette pratique permet de réduire l'impact de la latence et du débit sur le client pendant l'opération de repartitionnement. Elle permet aussi d'exécuter le repartitionnement plus rapidement, car un plus grand nombre de ressources peut être utilisé pour la redistribution des emplacements.

- Vérifiez le comportement hors délai du client – Certains clients peuvent observer une latence plus élevée lors d'un redimensionnement des clusters en ligne. La configuration de votre bibliothèque client avec un délai d'expiration supérieur peut être une aide en offrant au système le temps de se connecter même en cas de conditions de charge plus importantes sur le serveur. Dans certains cas, vous pouvez ouvrir un grand nombre de connexions sur le serveur. Dans ces cas, pensez à ajouter un backoff exponentiel à la logique de reconnexion. Cela peut empêcher qu'une rafale de nouvelles connexions atteignent le serveur simultanément.

Pendant le repartitionnement, appliquez les recommandations suivantes :

- Évitez les commandes onéreuses – Évitez d'exécuter des opérations gourmandes en calcul et en I/O, telles que les commandes `KEYS` et `SMEMBERS`. Nous suggérons cette approche, car ces opérations augmentent la charge sur le cluster et ont un impact sur ses performances. Utilisez à la place les commandes `SCAN` et `SSCAN`.
- Suivez les bonnes pratiques Lua – Évitez les longues exécutions de scripts Lua et déclarez toujours les clés utilisées dans les scripts Lua en amont. Nous recommandons cette approche pour déterminer que le script Lua n'utilise pas de commandes inter-emplacements. Veillez à ce que les clés utilisées dans les scripts Lua appartiennent au même emplacement.

Après le repartitionnement, notez ce qui suit :

- La diminution d'échelle peut être partiellement réussie si la mémoire sur les partitions cibles est insuffisante. Si un tel résultat se produit, vérifiez la mémoire disponible et réessayez l'opération, si nécessaire.
- Il n'est pas procédé à la migration des emplacements ayant des éléments volumineux. En particulier, les emplacements avec des éléments supérieurs à une post-sérialisation de 256 Mo ne font pas l'objet d'une migration.
- Les commandes `FLUSHALL` et `FLUSHDB` ne sont pas prises en charge dans les scripts Lua lors d’une opération de repartitionnement.

Comprendre la réplication MemoryDB

MemoryDB implémente la réplication avec des données partitionnées sur un maximum de 500 partitions.

Chaque partition d'un cluster compte un nœud simple primaire en lecture/écriture et jusqu'à 5 nœuds de réplica en lecture seule. Chaque nœud principal peut supporter jusqu'à 100 Mo/s. Vous pouvez créer un cluster avec un plus grand nombre de partitions et un nombre plus faible de répliques, pour un total de 500 nœuds par cluster. La configuration de ce cluster peut contenir de 500 partitions avec 0 réplica à 100 partitions avec 4 réplicas, ce qui correspond au nombre maximal de réplicas autorisé.

Cohérence

Dans MemoryDB, les nœuds principaux sont très cohérents. Les opérations d'écriture réussies sont stockées de manière durable dans des journaux transactionnels multi-AZ distribués avant d'être renvoyées aux clients. Les opérations de lecture sur les primaires renvoient toujours le plus de up-to-date données, reflétant les effets de toutes les opérations d'écriture précédentes réussies. Cette forte cohérence est préservée lors des basculements principaux.

Dans MemoryDB, les nœuds de réplica sont éventuellement cohérents. Les opérations de lecture à partir de répliques (à l'aide d'une `READONLY` commande) peuvent ne pas toujours refléter les effets des dernières opérations d'écriture réussies, les métriques de décalage étant publiées dans. CloudWatch Toutefois, les opérations de lecture à partir d'une seule réplique sont cohérentes de manière séquentielle. Les opérations d'écriture réussies prennent effet sur chaque réplique dans l'ordre dans lequel elles ont été exécutées sur la réplique principale.

Réplication dans un cluster

Chaque réplique de lecture d'une partition conserve une copie des données du nœud principal de la partition. Des mécanismes de réplication asynchrones utilisant les journaux de transactions sont utilisés pour maintenir la synchronisation des répliques de lecture avec la copie principale. Les applications peuvent lire à partir de n'importe quel nœud du cluster. Les applications ne peuvent écrire que sur les nœuds principaux. Les répliques de lecture améliorent l'évolutivité de la lecture. Comme MemoryDB stocke les données dans des journaux de transactions durables, il n'y a aucun risque de perte de données. Les données sont partitionnées entre les partitions d'un cluster MemoryDB.

Les applications utilisent le point de terminaison du cluster MemoryDB pour se connecter aux nœuds du cluster. Pour plus d'informations, veuillez consulter Recherche de points de terminaison de connexion.

Les clusters MemoryDB sont régionaux et ne peuvent contenir que des nœuds d'une région. Pour améliorer la tolérance aux pannes, vous devez provisionner des serveurs principaux et lire des répliques dans plusieurs zones de disponibilité au sein de cette région.

L'utilisation de la réplication, qui fournit la fonctionnalité Multi-AZ, est vivement recommandée pour tous les clusters MemoryDB. Pour plus d'informations, veuillez consulter Minimiser les temps d'arrêt dans MemoryDB avec Multi-AZ.

Minimiser les temps d'arrêt dans MemoryDB avec Multi-AZ

Dans un certain nombre de cas, MemoryDB peut avoir besoin de remplacer un nœud principal ; il s'agit notamment de certains types de maintenance planifiée et de l'éventualité peu probable d'une défaillance d'un nœud principal ou d'une zone de disponibilité.

La réponse à une défaillance d'un nœud dépend du nœud défaillant. Cependant, dans tous les cas, MemoryDB garantit qu'aucune donnée n'est perdue lors du remplacement ou du basculement des nœuds. Par exemple, si une réplique échoue, le nœud défaillant est remplacé et les données sont synchronisées à partir du journal des transactions. En cas de défaillance du nœud principal, un basculement est déclenché vers une réplique cohérente, ce qui garantit qu'aucune donnée n'est perdue pendant le basculement. Les écritures sont désormais effectuées depuis le nouveau nœud principal. L'ancien nœud principal est ensuite remplacé et synchronisé à partir du journal des transactions.

Si un nœud principal tombe en panne sur une partition à nœud unique (pas de répliques), MemoryDB cesse d'accepter les écritures jusqu'à ce que le nœud principal soit remplacé et synchronisé à partir du journal des transactions.

Le remplacement du nœud peut entraîner un certain temps d'arrêt pour le cluster, mais si Multi-AZ est actif, le temps d'arrêt est minimisé. Le rôle du nœud principal basculera automatiquement vers l'une des répliques. Il n'est pas nécessaire de créer et de provisionner un nouveau nœud principal, car MemoryDB gérera cela de manière transparente. Ce basculement et la promotion d'un réplica vous permettent de recommencer à écrire dans le nouveau nœud principal dès que la promotion est terminée.

Dans le cas de remplacements de nœuds planifiés initiés en raison de mises à jour de maintenance ou de mises à jour de service, sachez que les remplacements de nœuds prévus sont terminés pendant que le cluster traite les demandes d'écriture entrantes.

Le multi-AZ sur vos clusters MemoryDB améliore votre tolérance aux pannes. Cela est particulièrement vrai dans les cas où les nœuds principaux de votre cluster deviennent inaccessibles ou tombent en panne pour une raison quelconque. Le mode multi-AZ sur les clusters MemoryDB nécessite que chaque partition possède plusieurs nœuds et est automatiquement activé.

Rubriques

- Scénarios de défaillance avec réponses multi-AZ
- Test du basculement automatique

Scénarios de défaillance avec réponses multi-AZ

Si Multi-AZ est actif, un nœud principal défaillant bascule vers une réplique disponible. La réplique est automatiquement synchronisée avec le journal des transactions et devient principale, ce qui est beaucoup plus rapide que la création et le reprovisionnement d'un nouveau nœud principal. Ce processus dure généralement quelques secondes pendant lesquelles vous ne pouvez pas écrire sur le cluster.

Lorsque Multi-AZ est actif, MemoryDB surveille en permanence l'état du nœud principal. En cas de défaillance du nœud principal, l'une des actions suivantes est effectuée selon le type de la défaillance.

Rubriques

- Scénarios d'échec lorsque seul le nœud primaire échoue
- Scénarios de défaillance en cas de défaillance du nœud principal et de certaines répliques
- Scénarios d'échec lorsque l'ensemble du cluster tombe en panne

Scénarios d'échec lorsque seul le nœud primaire échoue

Si seul le nœud principal tombe en panne, une réplique deviendra automatiquement le nœud principal. Une réplique de remplacement est ensuite créée et mise en service dans la même zone de disponibilité que la réplique principale défaillante.

Lorsque seul le nœud principal tombe en panne, MemoryDB Multi-AZ effectue les opérations suivantes :

1. Le nœud principal défaillant est mis hors ligne.
2. Une up-to-date réplique devient automatiquement principale.

   Les écritures peuvent reprendre dès que le processus de basculement est terminé, généralement quelques secondes seulement.
3. Une réplique de remplacement est lancée et provisionnée.

   La réplique de remplacement est lancée dans la zone de disponibilité dans laquelle se trouvait le nœud principal défaillant afin que la distribution des nœuds soit maintenue.
4. La réplique est synchronisée avec le journal des transactions.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez les rubriques suivantes :

- Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Scénarios de défaillance en cas de défaillance du nœud principal et de certaines répliques

Si le principal et au moins un réplica échouent, un up-to-date réplica est promu en cluster principal. De nouvelles répliques sont également créées et approvisionnées dans les mêmes zones de disponibilité que les nœuds défaillants.

Lorsque le nœud principal et certaines répliques échouent, MemoryDB Multi-AZ effectue les opérations suivantes :

1. Le nœud principal défaillant et les répliques défaillantes sont mis hors ligne.
2. Une réplique disponible deviendra le nœud principal.

   Les écritures peuvent reprendre dès que le basculement est terminé, généralement quelques secondes seulement.
3. Des réplicas de remplacement sont créés et provisionnés.

   Les réplicas de remplacement sont créés dans les zones de disponibilité des nœuds ayant échoué afin que la distribution des nœuds soit maintenue.
4. Tous les nœuds sont synchronisés avec le journal des transactions.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez les rubriques suivantes :

- Trouver le point de terminaison d'un cluster MemoryDB (CLI)AWS
- Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Scénarios d'échec lorsque l'ensemble du cluster tombe en panne

En cas de défaillance générale, tous les nœuds sont recréés et mis en service dans les mêmes zones de disponibilité que les nœuds initiaux.

Il n'y a aucune perte de données dans ce scénario car les données étaient conservées dans le journal des transactions.

Lorsque l'ensemble du cluster échoue, MemoryDB Multi-AZ effectue les opérations suivantes :

1. Le nœud principal défaillant et les répliques sont mis hors ligne.
2. Un nœud principal de remplacement est créé et provisionné, synchronisé avec le journal des transactions.
3. Des répliques de remplacement sont créées et approvisionnées, synchronisées avec le journal des transactions.

   Les remplacements sont créés dans les zones de disponibilité des nœuds ayant échoué afin que la distribution des nœuds soit maintenue.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez les rubriques suivantes :

- Trouver le point de terminaison d'un cluster MemoryDB (CLI)AWS
- Trouver le point de terminaison d'un cluster MemoryDB (API MemoryDB)

Test du basculement automatique

Vous pouvez tester le basculement automatique à l'aide de la console MemoryDB, de l'API MemoryDB et de l' AWS CLI API MemoryDB.

Lors du test, tenez compte des points suivants :

- Vous pouvez utiliser cette opération jusqu'à cinq fois par période de 24 heures.
- Si vous appelez cette opération sur des partitions de différents clusters, vous pouvez effectuer les appels simultanément.
- Dans certains cas, vous pouvez appeler cette opération plusieurs fois sur différentes partitions du même cluster MemoryDB. Dans de tels cas, le premier remplacement de nœud doit se terminer avant qu'un appel ultérieur puisse être effectué.
- Pour déterminer si le remplacement du nœud est terminé, vérifiez les événements à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB. Recherchez les événements suivants liés à`FailoverShard`, listés ici par ordre d'occurrence probable :

  1. message du cluster : `FailoverShard API called for shard <shard-id>`
  2. message du cluster : `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  3. message du cluster : `Recovering nodes <node-id>`
  4. message du cluster : `Finished recovery for nodes <node-id>`

  Pour plus d’informations, consultez les ressources suivantes :

  - DescribeEventsdans la référence de l'API MemoryDB
- Cette API est conçue pour tester le comportement de votre application en cas de basculement de MemoryDB. Elle n’a pas été conçue pour être un outil opérationnel permettant de lancer un basculement pour résoudre un problème avec le cluster. De plus, dans certaines conditions, telles que des événements opérationnels à grande échelle, cette API AWS peut être bloquée.

Rubriques

- Test du basculement automatique à l'aide du AWS Management Console
- Test du basculement automatique à l'aide du AWS CLI
- Test du basculement automatique à l'aide de l'API MemoryDB

Test du basculement automatique à l'aide du AWS Management Console

Utilisez la procédure suivante pour tester le basculement automatique avec la console.

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Cliquez sur le bouton radio situé à gauche du cluster que vous souhaitez tester. Ce cluster doit comporter au moins un nœud de réplication.
3. Dans la zone Détails vérifiez que la fonctionnalité Multi-AZ est activée pour ce cluster. Si tel n'est pas le cas, choisissez un autre cluster ou modifiez-le afin d'activer Multi-AZ. Pour plus d’informations, consultez Modification d'un cluster MemoryDB.
4. Choisissez le nom du cluster.
5. Sur la page Partitions et nœuds, choisissez le nom de la partition sur laquelle vous souhaitez tester le basculement.
6. Pour le nœud, choisissez Failover Primary.
7. Choisissez Continuer pour basculer le nœud principal, ou sur Annuler pour annuler l'opération et ne pas basculer le nœud principal.

   Au cours du processus de basculement, la console continue à afficher le statut available du nœud. Pour suivre l'avancement du test de basculement, choisissezÉvénements dans le volet de navigation de la console. Sous l'onglet Événements, recherchez les événements indiquant que le basculement a commencé (`FailoverShard API called`) et est terminé (`Recovery completed`).

Test du basculement automatique à l'aide du AWS CLI

Vous pouvez tester le basculement automatique sur n'importe quel cluster compatible Multi-AZ à l'aide de l' AWS CLI opération failover-shard.

Paramètres

- `--cluster-name` : obligatoire. Le cluster qui doit être testé.
- `--shard-name` : obligatoire. Nom de la partition sur laquelle vous souhaitez tester le basculement automatique. Vous pouvez tester un maximum de cinq partitions sur une période continue de 24 heures.

L'exemple suivant utilise l'appel AWS CLI to `failover-shard` sur le shard `0001` du cluster MemoryDB. `my-cluster`

Pour Linux, macOS ou Unix :

```
aws memorydb failover-shard \
   --cluster-name my-cluster \
   --shard-name 0001
```

Pour Windows :

```
aws memorydb failover-shard ^
   --cluster-name my-cluster ^
   --shard-name 0001
```

Pour suivre la progression de votre basculement, utilisez l' AWS CLI `describe-events`opération.

Il renverra la réponse JSON suivante :

```
{
    "Events": [
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Failover to replica node my-cluster-0001-002 completed",
            "Date": "2021-08-22T12:39:37.568000-07:00"
        },
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Starting failover for shard 0001",
            "Date": "2021-08-22T12:39:10.173000-07:00"
        }
    ]
}
```

Pour plus d’informations, consultez les ressources suivantes :

- failover-shard
- describe-events

Test du basculement automatique à l'aide de l'API MemoryDB

L'exemple suivant fait `FailoverShard` appel à la partition `0003` du cluster`memorydb00`.

Example Test du basculement automatique

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=FailoverShard
    &ShardName=0003
    &ClusterName=memorydb00
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T192317Z
    &X-Amz-Credential=<credential>
```

Pour suivre la progression de votre basculement, utilisez l'opération d'API MemoryDB. `DescribeEvents`

Pour plus d’informations, consultez les ressources suivantes :

- FailoverShard
- DescribeEvents

Modification du nombre de réplicas

Vous pouvez augmenter ou diminuer dynamiquement le nombre de réplicas en lecture dans votre cluster MemoryDB à l'aide de laAWS Management Console, de la ou de l'API AWS CLI MemoryDB. Le nombre de réplicas doit contenir du même nombre de réplicas.

Augmentation du nombre de réplicas dans un cluster

Vous pouvez augmenter le nombre de réplicas dans un cluster MemoryDB jusqu'à un maximum de cinq réplicas par partition. Vous pouvez effectuer cette opération à l'aide de la AWS Management ConsoleAWS CLI, de la ou de l'API MemoryDB.

Rubriques

- Utilisation du AWS Management Console
- Utilisation du AWS CLI
- Utilisation de l'API MemoryDB

Utilisation du AWS Management Console

Pour augmenter le nombre de répliques dans un cluster MemoryDB (console), consultez. Ajouter/supprimer des nœuds d'un cluster

Utilisation du AWS CLI

Pour augmenter le nombre de réplicas dans un cluster MemoryDB, utilisez la `update-cluster` commande de l'avec les paramètres suivants :

- `--cluster-name` : obligatoire. Identifie le cluster dans lequel vous souhaitez augmenter le nombre de réplicas.
- `--replica-configuration` : obligatoire. Vous permet de définir le nombre de réplicas. Pour augmenter le nombre de réplicas, définissez la `ReplicaCount` propriété sur le nombre de réplicas que vous souhaitez dans cette partition à la fin de cette opération.

Example

L'exemple suivant augmente le nombre de réplicas dans le cluster `my-cluster` à 2 réplicas.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --replica-configuration \
        ReplicaCount=2
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=2
```

La réponse JSON suivante n'a pas d'importance :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour afficher les détails du cluster mis à jour une fois que son état passe de la phase de mise à jour à celle de disponibilité, utilisez la commande suivante :

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Il renverra la réponse JSON suivante :

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-003",
```

```
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-22T12:59:31.844000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 3
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```


Pour plus d'informations sur l'augmentation du nombre de réplicas à l'aide de la CLI, veuillez consulter update-cluster dans la Référence des commandes. AWS CLI

Utilisation de l'API MemoryDB

Pour augmenter le nombre de répliques dans un shard MemoryDB, utilisez l'`UpdateCluster`action avec les paramètres suivants :

- `ClusterName` : obligatoire. Identifie le cluster dans lequel vous souhaitez augmenter le nombre de réplicas.
- `ReplicaConfiguration` : obligatoire. Vous permet de définir le nombre de réplicas. Pour augmenter le nombre de réplicas, définissez la `ReplicaCount` propriété sur le nombre de réplicas que vous souhaitez dans cette partition à la fin de cette opération.

Example

L'exemple suivant augmente le nombre de réplicas dans le cluster `sample-cluster` à trois réplicas. Lorsque l'exemple est terminé, il y a trois répliques dans chaque shard. Ce numéro s'applique qu'il s'agisse d'un cluster MemoryDB avec une seule partition ou d'un cluster MemoryDB avec plusieurs partitions.

```
https://memory-db.us-east-1.amazonaws.com/
      ?Action=UpdateCluster
      &ReplicaConfiguration.ReplicaCount=3
      &ClusterName=sample-cluster
      &Version=2021-01-01
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20210802T192317Z
      &X-Amz-Credential=<credential>
```

Pour plus d'informations sur l'augmentation du nombre de réplicas à l'aide de l'interface de commande, veuillez consulter UpdateCluster.

Diminution du nombre de réplicas dans un cluster

Vous pouvez diminuer le nombre de réplicas dans un cluster pour MemoryDB. Vous pouvez diminuer le nombre de réplicas à 0, mais vous ne pouvez pas basculer vers une réplica si votre nœud principal tombe en panne.

Vous pouvez utiliserAWS Management Console, la AWS CLI ou l'API MemoryDB pour diminuer le nombre de réplicas dans un cluster.

Rubriques

- Utilisation du AWS Management Console
- Utilisation du AWS CLI
- Utilisation de l'API MemoryDB

Utilisation du AWS Management Console

Pour réduire le nombre de répliques dans un cluster MemoryDB (console), consultez. Ajouter/supprimer des nœuds d'un cluster

Utilisation du AWS CLI

Pour diminuer le nombre de réplicas dans un cluster MemoryDB, utilisez la `update-cluster` commande de l'avec les paramètres suivants :

- `--cluster-name` : obligatoire. Identifie le cluster dans lequel vous souhaitez réduire le nombre de répliques.
- `--replica-configuration` : obligatoire.

  `ReplicaCount`— Définissez cette propriété pour spécifier le nombre de nœuds de réplica souhaité.

Example

L'exemple suivant permet `--replica-configuration` de diminuer le nombre de réplicas dans le cluster `my-cluster` à la valeur spécifiée.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
```

```
    --cluster-name my-cluster \
    --replica-configuration \
        ReplicaCount=1
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=1 ^
```

Il renverra la réponse JSON suivante :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour afficher les détails du cluster mis à jour une fois que son état passe de la phase de mise à jour à celle de disponibilité, utilisez la commande suivante :

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Il renverra la réponse JSON suivante :

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
```

```
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```


Pour plus d'informations sur la diminution du nombre de réplicas à l'aide de la CLI, veuillez consulter update-cluster dans la Référence des commandes. AWS CLI

Utilisation de l'API MemoryDB

Pour réduire le nombre de répliques dans un cluster MemoryDB, utilisez l'`UpdateCluster`action avec les paramètres suivants :

- `ClusterName` : obligatoire. Identifie le cluster dans lequel vous souhaitez réduire le nombre de répliques.
- `ReplicaConfiguration` : obligatoire. Vous permet de définir le nombre de réplicas.

`ReplicaCount`— Définissez cette propriété pour spécifier le nombre de nœuds de réplica souhaité.

Example

L'exemple suivant permet `ReplicaCount` de diminuer le nombre de réplicas dans le cluster `sample-cluster` à une. Lorsque l'exemple est terminé, il y a une réplique dans chaque shard. Ce numéro s'applique qu'il s'agisse d'un cluster MemoryDB avec une seule partition ou d'un cluster MemoryDB avec plusieurs partitions.

```
https://memory-db.us-east-1.amazonaws.com/
      ?Action=UpdateCluster
      &ReplicaConfiguration.ReplicaCount=1
      &ClusterName=sample-cluster
      &Version=2021-01-01
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20210802T192317Z
      &X-Amz-Credential=<credential>
```

Pour plus d'informations sur la diminution du nombre de réplicas à l'aide de l'interface de commande, veuillez consulter UpdateCluster.

Instantané et restauration

Les clusters MemoryDB sauvegardent automatiquement les données dans un journal transactionnel multi-AZ, mais vous pouvez choisir de créer des point-in-time instantanés d'un cluster périodiquement ou à la demande. Ces instantanés peuvent être utilisés pour recréer un cluster à un point précédent ou pour créer un tout nouveau cluster. L'instantané comprend les métadonnées du cluster, ainsi que toutes les données du cluster. Tous les instantanés sont écrits sur Amazon Simple Storage Service (Amazon S3), qui fournit un stockage durable. À tout moment, vous pouvez restaurer vos données en créant un nouveau cluster MemoryDB et en le remplissant avec les données d'un instantané. Avec MemoryDB, vous pouvez gérer les instantanés à l'aide des API AWS Management Console, the AWS Command Line Interface (AWS CLI) et MemoryDB.

Rubriques

- Contraintes liées aux captures

- Coûts des instantanés
- Planification de snapshots automatiques
- Création d'instantanés manuels
- Création d'un instantané final
- Décrire les instantanés
- Copie d'un instantané
- Exportation d'un instantané
- Restaurer à partir d'un instantané
- Création d'un nouveau cluster avec un instantané créé en externe
- Marquage des instantanés
- Suppression d'un instantané

Contraintes liées aux captures

Tenez compte des contraintes suivantes lors de la planification ou de la création de clichés :

- Pour les clusters MemoryDB, la capture instantanée et la restauration sont disponibles pour tous les types de nœuds pris en charge.
- Au cours d'une période continue de 24 heures, vous ne pouvez pas créer plus de 20 instantanés manuels par cluster.
- MemoryDB prend uniquement en charge la prise de snapshots au niveau du cluster. MemoryDB ne prend pas en charge la prise de snapshots au niveau de la partition ou du nœud.
- Pendant le processus de capture instantanée, vous ne pouvez exécuter aucune autre opération d'API ou de CLI sur le cluster.
- Si vous supprimez un cluster et demandez un instantané final, MemoryDB prend toujours le cliché à partir des nœuds principaux. Cela garantit que vous capturez les toutes dernières données avant que le cluster ne soit supprimé.

Coûts des instantanés

Avec MemoryDB, vous pouvez stocker gratuitement un instantané pour chaque cluster MemoryDB actif. L'espace de stockage pour les instantanés supplémentaires est facturé au taux de 0,085 $/Go par mois pour toutes les régions. AWS Aucuns frais de transfert de données ne sont facturés

pour la création d'un instantané ou pour la restauration des données d'un instantané vers un cluster MemoryDB.

Planification de snapshots automatiques

Pour n'importe quel cluster MemoryDB, vous pouvez activer les instantanés automatiques. Lorsque les instantanés automatiques sont activés, MemoryDB crée quotidiennement un instantané du cluster. Il n'y a aucun impact sur le cluster et le changement est immédiat. Pour plus d'informations, consultez Restaurer à partir d'un instantané.

Lorsque vous planifiez des instantanés automatiques, vous devez planifier les paramètres suivants :

- Fenêtre de capture d'écran : période de chaque jour pendant laquelle MemoryDB commence à créer un instantané. La durée minimale de la fenêtre de capture d'écran est de 60 minutes. Vous pouvez définir la fenêtre des instantanés au moment qui vous convient le mieux ou à un moment de la journée qui évite de créer des instantanés pendant les périodes d'utilisation particulièrement intense.

  Si vous ne spécifiez pas de fenêtre de capture instantanée, MemoryDB en assigne une automatiquement.
- Limite de conservation des instantanés : nombre de jours pendant lesquels les instantanés sont conservés dans Amazon S3. Par exemple, si vous définissez la limite de rétention sur 5, un instantané pris aujourd'hui est conservé pendant 5 jours. Lorsque la limite de rétention expire, le cliché est automatiquement supprimé.

  La durée maximale de conservation des instantanés est de 35 jours. Si la limite de conservation des instantanés est définie sur 0, les instantanés automatiques sont désactivés pour le cluster. Les données MemoryDB sont toujours totalement durables même si la capture automatique des instantanés est désactivée.

Vous pouvez activer ou désactiver les instantanés automatiques lors de la création d'un cluster MemoryDB à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB. Vous pouvez activer les instantanés automatiques lorsque vous créez un cluster MemoryDB en cochant la case Activer les sauvegardes automatiques dans la section Instantanés. Pour plus d'informations, consultez Création d'un cluster MemoryDB.

Création d'instantanés manuels

Outre les instantanés automatiques, vous pouvez créer un instantané manuel à tout moment. Contrairement aux instantanés automatiques, qui sont automatiquement supprimés après une période de conservation spécifiée, les instantanés manuels n'ont pas de période de conservation après laquelle ils sont automatiquement supprimés. Vous devez supprimer manuellement tout instantané manuel. Même si vous supprimez un cluster ou un nœud, tous les instantanés manuels de ce cluster ou nœud sont conservés. Si vous ne souhaitez plus conserver un instantané manuel, vous devez le supprimer vous-même de manière explicite.

Les instantanés manuels sont utiles pour les tests et l'archivage. Par exemple, supposons que vous ayez développé un ensemble de données de base pour effectuer des tests. Vous pouvez créer un instantané manuel des données et le restaurer quand vous le souhaitez. Après avoir testé une application qui modifie les données, vous pouvez réinitialiser les données en créant un nouveau cluster et en effectuant une restauration à partir de votre instantané de référence. Lorsque le cluster est prêt, vous pouvez tester vos applications par rapport aux données de base à nouveau et répétez ce processus aussi souvent que nécessaire.

Outre la création directe d'un instantané manuel, vous pouvez créer un instantané manuel de l'une des manières suivantes :

- Copie d'un instantané— Peu importe que l'instantané source ait été créé automatiquement ou manuellement.
- Création d'un instantané final— Créez un instantané immédiatement avant de supprimer un cluster.

Autres sujets importants

- Contraintes liées aux captures
- Coûts des instantanés

Vous pouvez créer un instantané manuel d'un nœud à l'aide de l'API AWS Management Console, de AWS CLI, ou de l'API MemoryDB.

Création d'un instantané manuel (console)

Pour créer un instantané d'un cluster (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. dans le volet de navigation de gauche, choisissez Clusters.

   L'écran des clusters MemoryDB apparaît.
3. cliquez sur le bouton radio situé à gauche du nom du cluster MemoryDB que vous souhaitez sauvegarder.
4. Choisissez Actions, puis Prendre un instantané.
5. Dans la fenêtre Snapshot, saisissez le nom de votre instantané dans le champ Snapshot Name. Nous recommandons que le nom indique quel cluster a été sauvegardé, ainsi que la date et l'heure de création de l'instantané.

   Les contraintes d'attribution de noms de cluster sont les suivantes :

   - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
   - Doit commencer par une lettre.
   - Ils ne peuvent pas comporter deux traits d'union consécutifs.
   - Ils ne peuvent pas se terminer par un trait d'union.
6. Sous Chiffrement, choisissez d'utiliser une clé de chiffrement par défaut ou une clé gérée par le client. Pour plus d’informations, consultez Chiffrement en transit (TLS) dans MemoryDB.
7. Sous Balises, ajoutez éventuellement des balises pour rechercher et filtrer vos instantanés ou suivre vos AWS coûts.
8. Choisissez Prendre un instantané.

   L'état du cluster devient snapshotting. Lorsque le statut redevient disponible, le cliché est terminé.

Création d'un instantané manuel (AWS CLI)

Pour créer un instantané manuel d'un cluster à l'aide de AWS CLI, utilisez l'`create-snapshot` AWS CLI opération avec les paramètres suivants :

- `--cluster-name`— Nom du cluster MemoryDB à utiliser comme source pour le snapshot. Utilisez ce paramètre lors de la sauvegarde d'un cluster MemoryDB.

  Les contraintes d'attribution de noms de cluster sont les suivantes :

  - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
  - Doit commencer par une lettre.
  - Ils ne peuvent pas comporter deux traits d'union consécutifs.
  - Ils ne peuvent pas se terminer par un trait d'union.

- `--snapshot-name` – Nom de l'instantané à créer.

Rubriques en relation

Pour plus d’informations, consultez la section `create-snapshot` dans la référence des commandes AWS CLI .

Création d'un instantané manuel (API MemoryDB)

Pour créer un instantané manuel d'un cluster à l'aide de l'API MemoryDB, utilisez l'opération de l'API `CreateSnapshot` MemoryDB avec les paramètres suivants :

- `ClusterName`— Nom du cluster MemoryDB à utiliser comme source pour le snapshot. Utilisez ce paramètre lors de la sauvegarde d'un cluster MemoryDB.

  Les contraintes d'attribution de noms de cluster sont les suivantes :

  - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
  - Doit commencer par une lettre.
  - Ils ne peuvent pas comporter deux traits d'union consécutifs.
  - Ils ne peuvent pas se terminer par un trait d'union.
- `SnapshotName` – Nom de l'instantané à créer.

Rubriques en relation

Pour plus d'informations, consultez CreateSnapshot.

Création d'un instantané final

Vous pouvez créer un instantané final à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Création d'un instantané final (console)

Vous pouvez créer un instantané final lorsque vous supprimez un cluster MemoryDB à l'aide de la console MemoryDB.

Pour créer un instantané final lors de la suppression d'un cluster MemoryDB, sur la page de suppression, choisissez Oui et nommez l'instantané à. Étape 4 : Supprimer un cluster

Création d'un instantané final (AWS CLI)

Vous pouvez créer un instantané final lors de la suppression d'un cluster MemoryDB à l'aide du. AWS CLI

Lors de la suppression d'un cluster MemoryDB

Pour créer un instantané final lors de la suppression d'un cluster, utilisez l'`delete-cluster` AWS CLI opération, avec les paramètres suivants :

- `--cluster-name` – Nom du cluster en cours de suppression.
- `--final-snapshot-name`— Nom de l'instantané final.

Le code suivant prend le cliché final `bkup-20210515-final` lors de la suppression du clustermy`Cluster`.

Pour Linux, macOS ou Unix :

```
aws memorydb delete-cluster \
        --cluster-name myCluster \
        --final-snapshot-name bkup-20210515-final
```

Pour Windows :

```
aws memorydb delete-cluster ^
        --cluster-name myCluster ^
        --final-snapshot-name bkup-20210515-final
```

Pour plus d'informations, voir delete-cluster dans la référence des AWS CLI commandes.

Création d'un instantané final (API MemoryDB)

Vous pouvez créer un instantané final lors de la suppression d'un cluster MemoryDB à l'aide de l'API MemoryDB.

Lors de la suppression d'un cluster MemoryDB

Pour créer un instantané final, utilisez l'opération d'API `DeleteCluster` MemoryDB avec les paramètres suivants.

- `ClusterName` – Nom du cluster en cours de suppression.
- `FinalSnapshotName`— Nom de l'instantané.

L'opération d'API MemoryDB suivante crée le snapshot `bkup-20210515-final` lors de la suppression du cluster. `myCluster`

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteCluster
    &ClusterName=myCluster
    &FinalSnapshotName=bkup-20210515-final
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210515T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DeleteCluster.

Décrire les instantanés

Les procédures suivantes vous montrent comment afficher la liste de vos instantanés. Si vous le souhaitez, vous pouvez également afficher les détails d'un instantané en particulier.

Décrire les instantanés (console)

Pour afficher des instantanés à l'aide du AWS Management Console

1. Connectez-vous à la console
2. dans le volet de navigation de gauche, choisissez Snapshots.
3. Utilisez la recherche pour filtrer les instantanés manuels, automatiques ou tous les instantanés.
4. Pour voir les détails d'un instantané en particulier, cliquez sur le bouton radio situé à gauche du nom de l'instantané. Choisissez Actions, puis Afficher les détails.
5. Sur la page Afficher les détails, vous pouvez éventuellement effectuer des actions de capture d'écran supplémentaires, telles que copier, restaurer ou supprimer. Vous pouvez également ajouter des balises à l'instantané

Décrire les instantanés (AWS CLI)

Pour afficher la liste des instantanés et éventuellement les détails d'un instantané spécifique, utilisez l'opération `describe-snapshots` CLI.

Exemples

L'opération suivante utilise le paramètre `--max-results` pour répertorier jusqu'à 20 instantanés associés à votre compte. L'omission du paramètre permet de `--max-results` répertorier jusqu'à 50 instantanés.

```
aws memorydb describe-snapshots --max-results 20
```

L'opération suivante utilise le paramètre `--cluster-name` pour répertorier uniquement les instantanés associés au clustermy`-cluster`.

```
aws memorydb describe-snapshots --cluster-name my-cluster
```

L'opération suivante utilise le paramètre `--snapshot-name` pour afficher les détails de l'instantanémy`-snapshot`.

```
aws memorydb describe-snapshots --snapshot-name my-snapshot
```

Pour plus d'informations, consultez la section describe-snapshots.

Décrire les instantanés (API MemoryDB)

Pour afficher une liste d'instantanés, utilisez l'`DescribeSnapshots`opération.

Exemples

L'opération suivante utilise le paramètre `MaxResults` pour répertorier jusqu'à 20 instantanés associés à votre compte. L'omission du paramètre permet de `MaxResults` répertorier jusqu'à 50 instantanés.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &MaxResults=20
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

L'opération suivante utilise le paramètre `ClusterName` pour répertorier tous les instantanés associés au cluster`MyCluster`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &ClusterName=MyCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
```

```
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

L'opération suivante utilise le paramètre `SnapshotName` pour afficher les détails de l'instantané`MyBackup`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SnapshotName=MyBackup
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez DescribeSnapshots.

Copie d'un instantané

Vous pouvez faire une copie de n'importe quel instantané, qu'il ait été créé automatiquement ou manuellement. Lorsque vous copiez un instantané, la même clé de chiffrement KMS que la source est utilisée pour la cible, sauf si elle est spécifiquement remplacée. Vous pouvez également exporter votre instantané afin d'y accéder depuis l'extérieur de MemoryDB. Pour obtenir des conseils sur l'exportation de votre instantané, consultezExportation d'un instantané.

Les procédures suivantes indiquent comment copier un instantané.

Copier un instantané (console)

Pour copier un instantané (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Pour afficher la liste de vos instantanés, dans le volet de navigation de gauche, sélectionnez Snapshots.
3. Dans la liste des instantanés, cliquez sur le bouton radio situé à gauche du nom de l'instantané que vous souhaitez copier.
4. Choisissez Actions, puis Copier.
5. Sur la page Copier un instantané, procédez comme suit :

    a. Dans le champ Nom du nouvel instantané, tapez le nom de votre nouvel instantané.

    b. Ne remplissez pas la boîte Target S3 Bucket facultative. Ce champ ne doit être utilisé que pour exporter votre instantané et nécessite des autorisations S3 spéciales. Pour plus d'informations sur l'exportation d'un instantané, consultezExportation d'un instantané.

    c. Choisissez d'utiliser la clé de AWS KMS chiffrement par défaut ou une clé personnalisée. Pour plus d’informations, consultez Chiffrement en transit (TLS) dans MemoryDB.

    d. Vous pouvez éventuellement ajouter des balises à la copie instantanée.

    e. Choisissez Copier.

Copier un instantané (AWS CLI)

Pour copier un instantané, utilisez l'`copy-snapshot`opération.

Paramètres

- `--source-snapshot-name`— Nom de l'instantané à copier.
- `--target-snapshot-name`— Nom de la copie de l'instantané.
- `--target-bucket`— Réservé à l'exportation d'un instantané. N'utilisez pas ce paramètre lorsque vous effectuez une copie d'un instantané. Pour plus d'informations, consultez Exportation d'un instantané.

L'exemple suivant crée une copie d'un instantané automatique.

Pour Linux, macOS ou Unix :

```
aws memorydb copy-snapshot \
    --source-snapshot-name automatic.my-primary-2021-03-27-03-15 \
    --target-snapshot-name my-snapshot-copy
```

Pour Windows :

```
aws memorydb copy-snapshot ^
    --source-snapshot-name automatic.my-primary-2021-03-27-03-15 ^
    --target-snapshot-name my-snapshot-copy
```

Pour plus d'informations, consultez la section copy-snapshot.

Copier un instantané (API MemoryDB)

Pour copier un instantané, utilisez l'`copy-snapshot`opération avec les paramètres suivants :

Paramètres

- `SourceSnapshotName`— Nom de l'instantané à copier.
- `TargetSnapshotName`— Nom de la copie de l'instantané.
- `TargetBucket`— Réservé à l'exportation d'un instantané. N'utilisez pas ce paramètre lorsque vous effectuez une copie d'un instantané. Pour plus d'informations, consultez Exportation d'un instantané.

L'exemple suivant crée une copie d'un instantané automatique.

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=CopySnapshot
    &SourceSnapshotName=automatic.my-primary-2021-03-27-03-15
    &TargetSnapshotName=my-snapshot-copy
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez CopySnapshot.

Exportation d'un instantané

MemoryDB prend en charge l'exportation de votre instantané MemoryDB vers un bucket Amazon Simple Storage Service (Amazon S3), qui vous permet d'y accéder depuis l'extérieur de MemoryDB. Les instantanés MemoryDB exportés sont entièrement compatibles avec Redis OSS open source et peuvent être chargés avec la version ou l'outil Redis OSS approprié. Vous pouvez exporter un instantané à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

L'exportation d'un instantané peut s'avérer utile si vous devez lancer un cluster dans une autre AWS région. Vous pouvez exporter vos données dans une AWS région, copier le fichier .rdb dans la nouvelle AWS région, puis utiliser ce fichier .rdb pour amorcer le nouveau cluster au lieu d'attendre que le nouveau cluster soit renseigné par le biais de l'utilisation. Pour de plus amples informations sur l'amorçage d'un nouveau cluster, veuillez consulter Création d'un nouveau cluster avec un instantané créé en externe. Vous voulez aussi peut-être exporter les données de votre cluster pour utiliser le fichier .rdb lors d'un processus hors ligne.

Important

- L'instantané MemoryDB et le compartiment Amazon S3 dans lequel vous souhaitez le copier doivent se trouver dans la même AWS région.

  Bien que les instantanés copiés dans un compartiment Amazon S3 soient chiffrés, nous vous recommandons vivement de ne pas autoriser d'autres personnes à accéder au compartiment Amazon S3 dans lequel vous souhaitez stocker vos instantanés.
- L'exportation d'un instantané vers Amazon S3 n'est pas prise en charge pour les clusters utilisant la hiérarchisation des données. Pour plus d'informations, consultez Mise à niveau des données.

Avant de pouvoir exporter un instantané vers un compartiment Amazon S3, vous devez disposer d'un compartiment Amazon S3 dans la même AWS région que l'instantané. Accordez à MemoryDB l'accès au bucket. Les deux premières étapes vous indiquent comment procéder.

Warning

Dans les scénarios suivants, l'exposition de vos données peut ne pas vous convenir :

- Lorsqu'une autre personne a accès au compartiment Amazon S3 vers lequel vous avez exporté votre instantané.

  Pour contrôler l'accès à vos instantanés, autorisez uniquement l'accès au compartiment Amazon S3 aux personnes auxquelles vous souhaitez accéder à vos données. Pour plus d'informations sur la gestion de l'accès aux compartiments Amazon S3, veuillez consulter Contrôle d'accès dans le Guide du développeur Amazon S3.
- Lorsqu'une autre personne est autorisée à utiliser l'opération CopySnapshot API.

  Les utilisateurs ou les groupes autorisés à utiliser l'opération d'`CopySnapshot`API peuvent créer leurs propres compartiments Amazon S3 et y copier des instantanés. Pour contrôler l'accès à vos instantanés, utilisez une politique AWS Identity and Access Management (IAM) pour contrôler qui est autorisé à utiliser l'`CopySnapshot`API. Pour plus d'informations sur l'utilisation d'IAM pour contrôler l'utilisation des opérations de l'API MemoryDB, consultez Gestion des identités et des accès dans MemoryDB le guide de l'utilisateur de MemoryDB.

Rubriques

- Étape 1 : Créer un compartiment Amazon S3
- Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3
- Étape 3 : Exporter un instantané MemoryDB

Étape 1 : Créer un compartiment Amazon S3

La procédure suivante utilise la console Amazon S3 pour créer un compartiment Amazon S3 dans lequel vous exportez et stockez votre instantané MemoryDB.

Pour créer un compartiment 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. Choisissez Créer un compartiment.
3. Dans la fenêtre Create a Bucket – Select a Bucket Name and Region, procédez comme suit :

a. Dans Bucket Name (Nom du compartiment), indiquez le nom de votre compartiment Amazon S3.

b. Dans la liste des régions, choisissez une AWS région pour votre compartiment Amazon S3. Cette AWS région doit être la même AWS que l'instantané MemoryDB que vous souhaitez exporter.

c. Choisissez Créer.

Pour plus d'informations sur la création d'un compartiment Amazon S3, veuillez consulter Créer un compartiment dans le Guide de l'utilisateur Amazon Simple Storage Service.

Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3

AWS Les régions introduites avant le 20 mars 2019 sont activées par défaut. Vous pouvez commencer à travailler dans ces AWS régions immédiatement. Les régions introduites après le 20 mars 2019 sont désactivées par défaut. Vous devez activer ou adhérer à ces régions avant de pouvoir les utiliser, comme décrit dans Gestion des AWS régions.

Accorder à MemoryDB l'accès à votre compartiment S3 dans une région AWS

Pour créer les autorisations appropriées sur un compartiment Amazon S3 dans une AWS région, procédez comme suit.

Pour accorder à MemoryDB l'accès à un compartiment 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. Choisissez le nom du compartiment Amazon S3 dans lequel vous souhaitez copier le snapshot. Il doit s'agir du compartiment S3 que vous avez créé dans Étape 1 : Créer un compartiment Amazon S3.
3. Choisissez l'onglet Permissions et sous Permissions, choisissez Bucket policy.
4. Mettez à jour la politique pour accorder à MemoryDB les autorisations requises pour effectuer des opérations :

   - Ajoutez `[ "Service" : "region-full-name.memorydb-snapshot.amazonaws.com" ]` à `Principal`.
   - Ajoutez les autorisations suivantes requises pour exporter un instantané vers le compartiment Amazon S3.

- "s3:PutObject"
- "s3:GetObject"
- "s3:ListBucket"
- "s3:GetBucketAcl"
- "s3:ListMultipartUploadParts"
- "s3:ListBucketMultipartUploads"

La politique mise à jour devrait ressembler à l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "aws-region.memorydb-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::example-bucket",
                "arn:aws:s3:::example-bucket/*"
            ]
        }
    ]
}
```

Étape 3 : Exporter un instantané MemoryDB

Vous avez maintenant créé votre compartiment S3 et accordé à MemoryDB les autorisations nécessaires pour y accéder. Modifiez la propriété de l'objet S3 pour activer les ACL (propriétaire du

compartiment préféré). Ensuite, vous pouvez utiliser la console MemoryDB, la AWS CLI ou l'API MemoryDB pour y exporter votre instantané. La procédure suivante suppose que vous disposez des autorisations IAM suivantes spécifiques à S3.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "s3:GetBucketLocation",
   "s3:ListAllMyBuckets",
   "s3:PutObject",
   "s3:GetObject",
   "s3:DeleteObject",
   "s3:ListBucket"
  ],
  "Resource": "arn:aws:s3:::*"
 }]
}
```

Exportation d'un instantané MemoryDB (console)

Le processus suivant utilise la console MemoryDB pour exporter un instantané vers un compartiment Amazon S3 afin que vous puissiez y accéder depuis l'extérieur de MemoryDB. Le compartiment Amazon S3 doit se trouver dans la même AWS région que le snapshot MemoryDB.

Pour exporter un instantané MemoryDB vers un compartiment Amazon S3

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Pour afficher la liste de vos instantanés, dans le volet de navigation de gauche, sélectionnez Snapshots.
3. Dans la liste des instantanés, cliquez sur le bouton radio situé à gauche du nom de l'instantané que vous souhaitez exporter.
4. Choisissez Copier.
5. Dans Create Copy of the Backup? (Créer une copie de la sauvegarde ?), procédez comme suit :

   a. Dans le champ Nom du nouvel instantané, tapez le nom de votre nouvel instantané.

      Le nom doit comprendre entre 1 et 1 000 caractères et pouvoir être encodé en UTF-8.

MemoryDB ajoute un identifiant de partition et `.rdb` à la valeur que vous entrez ici. Par exemple, si vous entrez`my-exported-snapshot`, MemoryDB crée. `my-exported-snapshot-0001.rdb`

b. Dans la liste des emplacements S3 cibles, choisissez le nom du compartiment Amazon S3 dans lequel vous souhaitez copier votre instantané (le compartiment dans lequel vous l'avez créé[Étape 1 : Créer un compartiment Amazon S3]).

   L'emplacement S3 cible doit être un compartiment Amazon S3 situé dans la AWS région du snapshot avec les autorisations suivantes pour que le processus d'exportation réussisse.

   - Accès à l'objet – Lecture et Écriture.
   - Accès aux autorisations – Lecture.

   Pour plus d'informations, consultez [Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3].

c. Choisissez Copier.

> **Note**
>
> Si votre compartiment S3 ne dispose pas des autorisations nécessaires pour que MemoryDB puisse y exporter un instantané, vous recevez l'un des messages d'erreur suivants. Retournez [Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3] à pour ajouter les autorisations spécifiées et réessayez d'exporter votre instantané.
>
> - MemoryDB n'a pas obtenu les autorisations READ %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations de lecture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations WRITE %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations d'écriture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations READ_ACP %s sur le compartiment S3.
>
>   Solution : ajoutez Read pour l'accès aux autorisations sur le compartiment.

Si vous souhaitez copier votre instantané AWS dans une autre région, utilisez Amazon S3 pour le copier. Pour plus d'informations, consultez Copier des objets dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Exportation d'un instantané MemoryDB (CLI)AWS

Exportez le snapshot vers un compartiment Amazon S3 à l'aide de l'opération `copy-snapshot` CLI avec les paramètres suivants :

Paramètres

- `--source-snapshot-name`— Nom de l'instantané à copier.
- `--target-snapshot-name`— Nom de la copie de l'instantané.

  Le nom doit comprendre entre 1 et 1 000 caractères et pouvoir être encodé en UTF-8.

  MemoryDB ajoute un identifiant de partition et `.rdb` à la valeur que vous entrez ici. Par exemple, si vous entrez`my-exported-snapshot`, MemoryDB crée. `my-exported-snapshot-0001.rdb`
- `--target-bucket`— Nom du compartiment Amazon S3 dans lequel vous souhaitez exporter l'instantané. Une copie de l'instantané est créée dans le compartiment spécifié.

  `--target-bucket`Il doit s'agir d'un compartiment Amazon S3 situé dans la AWS région du snapshot avec les autorisations suivantes pour que le processus d'exportation réussisse.
  - Accès à l'objet – Lecture et Écriture.
  - Accès aux autorisations – Lecture.

  Pour plus d’informations, consultez Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3.

L'opération suivante copie un instantané dans my-s3-bucket.

Pour Linux, macOS ou Unix :

```
aws memorydb copy-snapshot \
    --source-snapshot-name automatic.my-primary-2021-06-27-03-15 \
    --target-snapshot-name my-exported-snapshot \
    --target-bucket my-s3-bucket
```

Pour Windows :

```
aws memorydb copy-snapshot ^
    --source-snapshot-name automatic.my-primary-2021-06-27-03-15 ^
    --target-snapshot-name my-exported-snapshot ^
    --target-bucket my-s3-bucket
```

> **Note**
>
> Si votre compartiment S3 ne dispose pas des autorisations nécessaires pour que MemoryDB puisse y exporter un instantané, vous recevez l'un des messages d'erreur suivants. Retournez Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3 à pour ajouter les autorisations spécifiées et réessayez d'exporter votre instantané.
>
> - MemoryDB n'a pas obtenu les autorisations READ %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations de lecture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations WRITE %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations d'écriture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations READ_ACP %s sur le compartiment S3.
>
>   Solution : ajoutez Read pour l'accès aux autorisations sur le compartiment.

Pour plus d'informations, consultez la section `copy-snapshot` dans la référence des commandes AWS CLI .

Si vous souhaitez copier votre instantané AWS dans une autre région, utilisez Amazon S3 copy. Pour plus d'informations, consultez Copier des objets dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Exportation d'un instantané MemoryDB (API MemoryDB)

Exportez le snapshot vers un compartiment Amazon S3 à l'aide de l'opération d'`CopySnapshot`API avec ces paramètres.

Paramètres

- `SourceSnapshotName`— Nom de l'instantané à copier.
- `TargetSnapshotName`— Nom de la copie de l'instantané.

Le nom doit comprendre entre 1 et 1 000 caractères et pouvoir être encodé en UTF-8.

MemoryDB ajoute un identifiant de partition et `.rdb` à la valeur que vous entrez ici. Par exemple, si vous entrez `my-exported-snapshot`, vous obtenez `my-exported-snapshot-0001.rdb`.

- `TargetBucket`— Nom du compartiment Amazon S3 dans lequel vous souhaitez exporter l'instantané. Une copie de l'instantané est créée dans le compartiment spécifié.

  `TargetBucket`Il doit s'agir d'un compartiment Amazon S3 situé dans la AWS région du snapshot avec les autorisations suivantes pour que le processus d'exportation réussisse.

  - Accès à l'objet – Lecture et Écriture.
  - Accès aux autorisations – Lecture.

  Pour plus d’informations, consultez Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3.

L'exemple suivant crée une copie d'un instantané automatique dans le compartiment Amazon S3`my-s3-bucket`.

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=CopySnapshot
    &SourceSnapshotName=automatic.my-primary-2021-06-27-03-15
    &TargetBucket=my-s3-bucket
    &TargetSnapshotName=my-snapshot-copy
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

> **Note**
>
> Si votre compartiment S3 ne dispose pas des autorisations nécessaires pour que MemoryDB puisse y exporter un instantané, vous recevez l'un des messages d'erreur suivants. Retournez Étape 2 : accorder à MemoryDB l'accès à votre compartiment Amazon S3 à pour ajouter les autorisations spécifiées et réessayez d'exporter votre instantané.
>
> - MemoryDB n'a pas obtenu les autorisations READ %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations de lecture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations WRITE %s sur le compartiment S3.
>
>   Solution : ajoutez des autorisations d'écriture sur le compartiment.
> - MemoryDB n'a pas obtenu les autorisations READ_ACP %s sur le compartiment S3.
>
>   Solution : ajoutez Read pour l'accès aux autorisations sur le compartiment.

Pour plus d'informations, consultez CopySnapshot.

Si vous souhaitez copier votre instantané AWS dans une autre région, utilisez Amazon S3 copy pour copier l'instantané exporté dans le compartiment Amazon S3 d'une autre AWS région. Pour plus d'informations, consultez Copier des objets dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Restaurer à partir d'un instantané

Vous pouvez restaurer les données d'un fichier instantané MemoryDB ou ElastiCache (Redis OSS) .rdb vers un nouveau cluster à tout moment.

Le processus de restauration de MemoryDB prend en charge les éléments suivants :

- Migration d'un ou de plusieurs fichiers instantanés .rdb que vous avez créés ElastiCache (Redis OSS) vers un cluster MemoryDB.

  Les fichiers .rdb doivent être placés dans S3 pour que la restauration soit possible.
- Spécifier un nombre de partitions dans le nouveau cluster différent du nombre de partitions du cluster utilisé pour créer le fichier de capture instantanée.
- Spécification d'un type de nœud différent pour le nouveau cluster — plus grand ou plus petit. Si vous optez pour un type de nœud plus petit, assurez-vous que le nouveau type de nœud dispose de suffisamment de mémoire pour vos données et pour la surcharge de Redis OSS.
- Configuration des emplacements du nouveau cluster MemoryDB différemment de celle du cluster utilisé pour créer le fichier de capture instantanée.

> **Important**
>
> - Les clusters MemoryDB ne prennent pas en charge plusieurs bases de données. Par conséquent, lors de la restauration sur MemoryDB, votre restauration échoue si le fichier .rdb fait référence à plusieurs bases de données.
> - Vous ne pouvez pas restaurer un instantané d'un cluster qui utilise la hiérarchisation des données (par exemple, type de nœud r6gd) dans un cluster qui n'utilise pas la hiérarchisation des données (par exemple, type de nœud r6g).

La question de savoir si vous apportez des modifications lors de la restauration d'un cluster à partir d'un instantané dépend des choix que vous faites. Vous pouvez effectuer ces choix sur la page Restaurer le cluster lorsque vous utilisez la console MemoryDB pour effectuer une restauration. Vous pouvez effectuer ces choix en définissant des valeurs de paramètres lorsque vous utilisez l'API AWS CLI ou MemoryDB pour effectuer une restauration.

Au cours de l'opération de restauration, MemoryDB crée le nouveau cluster, puis le remplit avec les données du fichier de capture instantanée. Lorsque ce processus est terminé, le cluster est réchauffé et prêt à accepter les demandes.

> **Important**
>
> Avant de poursuivre, assurez-vous d'avoir créé un instantané du cluster à partir duquel vous souhaitez effectuer la restauration. Pour plus d'informations, consultez Création d'instantanés manuels.
> Si vous souhaitez effectuer une restauration à partir d'un instantané créé en externe, consultezCréation d'un nouveau cluster avec un instantané créé en externe.

Les procédures suivantes vous montrent comment restaurer un instantané sur un nouveau cluster à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Restauration à partir d'un instantané (console)

Pour restaurer un instantané sur un nouveau cluster (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation, sélectionnez Snapshots.
3. Dans la liste des instantanés, cliquez sur le bouton à côté du nom du cliché à partir duquel vous souhaitez effectuer la restauration.
4. Choisissez Actions, puis Restaurer
5. Dans Configuration du cluster, entrez ce qui suit :

    a. Nom du cluster : obligatoire. Nom du nouveau cluster.
    b. Description — Facultatif. Description du nouveau cluster.
6. Complétez la section Groupes de sous-réseaux :

    - Pour les groupes de sous-réseaux, créez un nouveau groupe de sous-réseaux ou choisissez-en un existant dans la liste disponible que vous souhaitez appliquer à ce cluster. Si vous en créez un nouveau :

        - Entrez un nom
        - Entrez une description

- Si vous avez activé Multi-AZ, le groupe de sous-réseaux doit contenir au moins deux sous-réseaux résidant dans des zones de disponibilité différentes. Pour plus d'informations, consultez Sous-réseaux et groupes de sous-réseaux.
- Si vous créez un nouveau groupe de sous-réseaux et que vous n'avez pas de VPC existant, il vous sera demandé de créer un VPC. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

7. Complétez la section Paramètres du cluster :

   a. Pour la compatibilité des versions de Redis OSS, acceptez la valeur par défaut`6.0`.

   b. Pour Port, acceptez le port Redis OSS par défaut 6379 ou, si vous avez une raison d'utiliser un autre port, entrez le numéro de port.

   c. Pour Groupe de paramètres, acceptez le groupe de `default.memorydb-redis6` paramètres.

      Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres spécifiques à Redis OSS.

   d. Pour Type de nœud, choisissez une valeur pour le type de nœud (ainsi que la taille de mémoire associée) que vous souhaitez.

      Si vous choisissez un membre de la famille de types de nœuds r6gd, vous activerez automatiquement la hiérarchisation des données dans votre cluster. Pour plus d'informations, consultez Mise à niveau des données.

   e. Dans Nombre de partitions, choisissez le nombre de partitions que vous souhaitez pour ce cluster.

      Vous pouvez modifier le nombre de partitions de votre cluster de manière dynamique. Pour plus d'informations, consultez Dimensionnement des clusters MemoryDB.

   f. Pour Réplicas par partition, choisissez le nombre de nœuds de réplica en lecture souhaité dans chaque partition.

      Les restrictions suivantes existent ;.

      - Si Multi-AZ est activé, assurez-vous d'avoir au moins un réplica par partition.
      - Le nombre de réplicas est le même pour chaque partition lors de la création du cluster à l'aide de la console.

g. Choisissez Next (Suivant)

h. Complétez la section Paramètres avancés :

i. Pour Groupes de sécurité, choisissez les groupes de sécurité que vous souhaitez utiliser pour ce cluster. Un groupe de sécurité agit comme un pare-feu pour contrôler l'accès réseau à votre cluster. Vous pouvez utiliser le groupe de sécurité par défaut pour votre VPC ou en créer un nouveau.

Pour plus d'informations sur les groupes de sécurité, consultez Groupes de sécurité pour votre VPC dans le Guide de l'utilisateur Amazon VPC.

ii. Les données sont cryptées de la manière suivante :

- Encryption at rest (Chiffrement au repos) : active le chiffrement des données stockées sur le disque. Pour de plus amples informations, veuillez consulter Chiffrement au repos.

> **Note**
>
> Vous avez la possibilité de fournir une autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé.

- Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Cela est activé par défaut. Pour de plus amples informations, veuillez consulter Chiffrement en transit.

Si vous ne sélectionnez aucun chiffrement, une liste de contrôle d'accès ouverte appelée « accès ouvert » sera créée avec un utilisateur par défaut. Pour plus d'informations, consultez Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL).

iii. Pour Snapshot, spécifiez éventuellement une période de conservation des instantanés et une fenêtre de capture d'écran. Par défaut, l'option Activer les instantanés automatiques est sélectionnée.

iv. Pour la fenêtre de maintenance, spécifiez éventuellement une fenêtre de maintenance. La fenêtre de maintenance est la période, généralement d'une heure, pendant laquelle MemoryDB planifie la maintenance du système pour votre cluster chaque semaine. Vous pouvez autoriser MemoryDB à choisir le jour et l'heure de votre fenêtre de maintenance (aucune préférence), ou vous pouvez choisir vous-même le jour, l'heure et

la durée (Spécifiez la fenêtre de maintenance). Si vous choisissez Specify maintenance window, choisissez dans les listes les valeurs de Start day, Start time et Duration (en heures) pour le créneau de maintenance. Toutes les heures sont en UTC.

Pour plus d'informations, consultez Gestion de la maintenance.

v. Pour Notifications, choisissez une rubrique Amazon Simple Notification Service (Amazon SNS) existante ou choisissez une entrée ARN manuelle et tapez l'Amazon Resource Name (ARN) de la rubrique. Amazon SNS permet d'émettre des notifications push vers des appareils connectés à Internet. La valeur par défaut consiste à désactiver les notifications. Pour plus d'informations, consultez https://aws.amazon.com/sns/.

i. Pour les tags, vous pouvez éventuellement appliquer des tags pour rechercher et filtrer vos clusters ou suivre vos AWS coûts.

j. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Créer un cluster pour lancer votre cluster ou Annuler pour annuler l'opération.

Dès que l'état de votre cluster est disponible, vous pouvez accorder un accès EC2, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 2 : Autoriser l'accès au cluster et Étape 3 : Connexion au cluster.

> **Important**
>
> Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter Étape 4 : Supprimer un cluster.

Restauration à partir d'un instantané (AWS CLI)

Lorsque vous utilisez l'une ou l'autre de ces `create-cluster` opérations, veillez à inclure le paramètre `--snapshot-name` ou `--snapshot-arns` à démarrer le nouveau cluster avec les données de l'instantané.

Pour plus d'informations, consultez les ressources suivantes :

- Création d'un cluster (AWS CLI)dans le guide de l'utilisateur de MemoryDB.

- create-cluster dans la référence des AWS CLI commandes.

Restauration à partir d'un instantané (API MemoryDB)

Vous pouvez restaurer un instantané MemoryDB à l'aide de l'opération d'API MemoryDB. `CreateCluster`

Lorsque vous utilisez `CreateCluster` cette opération, veillez à inclure le paramètre `SnapshotName` ou `SnapshotArns` à créer le nouveau cluster avec les données de l'instantané.

Pour plus d’informations, consultez les ressources suivantes :

- Création d'un cluster (API MemoryDB)dans le guide de l'utilisateur de MemoryDB.
- CreateClusterdans la référence de l'API MemoryDB.

Création d'un nouveau cluster avec un instantané créé en externe

Lorsque vous créez un nouveau cluster MemoryDB, vous pouvez l'amorcer avec les données d'un fichier instantané Redis OSS .rdb.

Pour amorcer un nouveau cluster MemoryDB à partir d'un instantané MemoryDB ou d'un instantané ElastiCache (Redis OSS), consultez. Restaurer à partir d'un instantané

Lorsque vous utilisez un fichier Redis OSS .rdb pour amorcer un nouveau cluster MemoryDB, vous pouvez effectuer les opérations suivantes :

- Spécifiez un certain nombre de partitions dans le nouveau cluster. Ce nombre peut être différent du nombre de partitions du cluster qui a été utilisé pour créer le fichier de capture instantanée.
- Spécifiez un type de nœud différent pour le nouveau cluster, plus grand ou plus petit que celui utilisé dans le cluster qui a créé le cliché. Si vous optez pour un type de nœud plus petit, assurez-vous que le nouveau type de nœud dispose de suffisamment de mémoire pour vos données et pour la surcharge de Redis OSS.

> **Important**
>
> - Vous devez vous assurer que les données de vos instantanés ne dépassent pas les ressources du nœud.
>
> Si le cliché est trop volumineux, le cluster obtenu a un statut de`restore-failed`. Si cela se produit, vous devez supprimer le cluster et recommencer.
>
> Pour une liste complète des types de nœuds et des spécifications, consultezParamètres spécifiques au type de nœud MemoryDB.
> - Vous pouvez chiffrer un fichier Redis OSS .rdb uniquement avec le chiffrement côté serveur Amazon S3 (SSE-S3). Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur.

Étape 1 : créer un instantané Redis OSS sur un cluster externe

Pour créer le snapshot destiné à démarrer votre cluster MemoryDB

1. Connectez-vous à votre instance Redis OSS existante.

2. Exécutez le Redis OSS `BGSAVE` ou `SAVE` l'opération pour créer un instantané. Notez l'emplacement de votre fichier .rdb.

   `BGSAVE` est asynchrone et ne bloque pas les autres clients lors du traitement. Pour plus d'informations, consultez BGSAVE sur le site Web de Redis OSS.

   `SAVE` est synchrone et bloque les autres processus jusqu'à la fin. Pour plus d'informations, consultez SAVE sur le site Web de Redis OSS.

Pour plus d'informations sur la création d'un instantané, consultez la section Persistance de Redis OSS sur le site Web de Redis OSS.

Étape 2 : Créer un compartiment et un dossier Amazon S3

Lorsque vous avez créé le fichier d'instantané, vous devez le télécharger dans un dossier au sein d'un compartiment Amazon S3. Pour cela, vous devez disposer d'un compartiment Amazon S3 et d'un dossier dans ce compartiment. Si vous avez déjà un compartiment et un dossier Amazon S3 avec les autorisations appropriées, vous pouvez ignorer cette étape et passer à Étape 3 : Chargez votre instantané sur Amazon S3.

Pour créer un compartiment 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. Pour créer un compartiment Amazon S3, suivez les instructions de la section Créer un compartiment dans le Guide de l'utilisateur de la console Amazon Simple Storage Service.

   Le nom de votre compartiment Amazon S3 doit être conforme au DNS. Dans le cas contraire, MemoryDB ne pourra pas accéder à votre fichier de sauvegarde. Les règles de conformité DNS sont les suivantes :

   - Les noms de compartiments doivent comporter entre 3 et 63 caractères.
   - Les noms doivent être une série d'une ou plusieurs étiquettes séparées par un point (.) où chaque étiquette :
     - Il doit commencer par une minuscule ou un chiffre.
     - Il doit terminer par une minuscule ou un chiffre.
     - Contient uniquement des lettres minuscules, des chiffres et des traits d'union.
   - Il ne peut pas présenter le même format qu'une adresse IP (par exemple, 192.0.2.0).

Nous vous recommandons vivement de créer votre compartiment Amazon S3 dans la même AWS région que votre nouveau cluster MemoryDB. Cette approche garantit la vitesse de transfert de données la plus élevée lorsque MemoryDB lit votre fichier .rdb depuis Amazon S3.

> **Note**
>
> Pour sécuriser au maximum vos données, définissez les autorisations les plus restrictives possible sur votre compartiment Amazon S3. Dans le même temps, les autorisations doivent toujours autoriser l'utilisation du bucket et de son contenu pour démarrer votre nouveau cluster MemoryDB.

Pour ajouter un dossier à un compartiment 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. Choisissez le nom du compartiment dans lequel le fichier .rdb sera téléchargé.
3. Choisissez Créer un dossier.
4. Saisissez un nom dans votre nouveau dossier.
5. Choisissez Enregistrer.

   Notez le nom du compartiment et celui du dossier.

Étape 3 : Chargez votre instantané sur Amazon S3

Maintenant, téléchargez le fichier .rdb que vous avez créé dans Étape 1 : créer un instantané Redis OSS sur un cluster externe. Vous le téléchargez dans le compartiment Amazon S3 et le dossier que vous avez créé dans Étape 2 : Créer un compartiment et un dossier Amazon S3. Pour plus d'informations sur cette tâche, consultez la section Chargement d'objets. Entre les étapes 2 et 3, choisissez le nom du dossier que vous avez créé.

Pour charger votre fichier .rdb dans un dossier 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. Choisissez le nom du compartiment Amazon S3 que vous avez créé à l'étape 2.

3. Choisissez le nom du dossier que vous avez créé à l'étape 2.
4. Sélectionnez Charger.
5. Choisissez Add files.
6. Recherchez le ou les fichiers que vous souhaitez charger, puis choisissez-les. Pour choisir plusieurs fichiers, maintenez la touche Ctrl enfoncée pendant que vous sélectionnez chaque nom de fichier.
7. Choisissez Ouvrir.
8. Vérifiez que le ou les fichiers corrects sont répertoriés sur la page de téléchargement, puis choisissez Charger.

Notez le chemin de votre fichier .rdb. Par exemple, si le nom de votre compartiment est `myBucket` et que le chemin est `myFolder/redis.rdb`, entrez `myBucket/myFolder/redis.rdb`. Vous avez besoin de ce chemin pour ensemencer le nouveau cluster avec les données de cet instantané.

Pour plus d'informations, consultez les règles de dénomination des compartiments dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Étape 4 : Accorder à MemoryDB l'accès en lecture au fichier .rdb

AWS Les régions introduites avant le 20 mars 2019 sont activées par défaut. Vous pouvez commencer à travailler dans ces AWS régions immédiatement. Les régions introduites après le 20 mars 2019 sont désactivées par défaut. Vous devez activer ou adhérer à ces régions avant de pouvoir les utiliser, comme décrit dans Gestion des AWS régions.

Accorder à MemoryDB un accès en lecture au fichier .rdb

Pour accorder à MemoryDB un accès en lecture au fichier instantané

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment S3 qui contient votre fichier .rdb.
3. Choisissez le nom du dossier qui contient votre fichier .rdb.
4. Choisissez le nom de votre fichier de capture d'écran .rdb. Le nom du fichier sélectionné apparaît au-dessus des onglets en haut de la page.
5. Choisissez l’onglet Permissions (Autorisations).

6. Sous Permissions (Autorisations), choisissez Bucket policy (Politique de compartiment), puis Edit (Modifier).
7. Mettez à jour la politique pour accorder à MemoryDB les autorisations requises pour effectuer des opérations :

   - Ajoutez `[ "Service" : "`*region-full-name*`.memorydb-snapshot.amazonaws.com" ]` à `Principal`.
   - Ajoutez les autorisations suivantes requises pour exporter un instantané vers le compartiment Amazon S3 :
     - `"s3:GetObject"`
     - `"s3:ListBucket"`
     - `"s3:GetBucketAcl"`

   La politique mise à jour devrait ressembler à l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "us-east-1.memorydb-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::example-bucket",
                "arn:aws:s3:::example-bucket/snapshot1.rdb",
                "arn:aws:s3:::example-bucket/snapshot2.rdb"
            ]
        }
    ]
}
```

8. Choisissez Enregistrer.

Étape 5 : amorcer le cluster MemoryDB avec les données du fichier .rdb

Vous êtes maintenant prêt à créer un cluster MemoryDB et à l'ensemencer avec les données du fichier .rdb. Pour créer le cluster, suivez les instructions surCréation d'un cluster MemoryDB.

La méthode que vous utilisez pour indiquer à MemoryDB où se trouve le snapshot Redis OSS que vous avez chargé sur Amazon S3 dépend de la méthode que vous avez utilisée pour créer le cluster :

Ensemencez le cluster MemoryDB avec les données du fichier .rdb

- Utilisation de la console MemoryDB

  Après avoir choisi le moteur Redis OSS, développez la section Paramètres avancés de Redis OSS et localisez Importer les données vers le cluster. Dans la zone Seed RDB file S3 location (Ensemencer l'emplacement S3 du fichier RDB), tapez le chemin d'accès Amazon S3 pour le ou les fichiers. Si vous avez plusieurs fichiers .rdb, tapez le chemin d'accès à chaque fichier dans une liste séparée par des virgules. Le chemin Amazon S3 ressemble à *myBucket*/*myFolder*/*myBackupFilename*.rdb.
- En utilisant le AWS CLI

  Si vous utilisez l'opération `create-cluster` ou `create-cluster`, définissez le paramètre `--snapshot-arns` afin de spécifier un ARN qualifié pour chaque fichier .rdb. Par exemple, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. L'ARN doit être résolu en fonction des fichiers de capture que vous avez stockés dans Amazon S3.
- Utilisation de l'API MemoryDB

  Si vous utilisez l'opération `CreateCluster` ou l'API `CreateCluster` MemoryDB, utilisez le paramètre `SnapshotArns` pour spécifier un ARN complet pour chaque fichier .rdb. Par exemple, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. L'ARN doit être résolu en fonction des fichiers de capture que vous avez stockés dans Amazon S3.

Au cours du processus de création de votre cluster, les données de votre instantané sont écrites dans le cluster. Vous pouvez suivre la progression en consultant les messages d'événements MemoryDB. Pour ce faire, consultez la console MemoryDB et choisissez Events. Vous pouvez également utiliser l'interface de ligne de commande AWS MemoryDB ou l'API MemoryDB pour obtenir des messages d'événements.

Marquage des instantanés

Vous pouvez attribuer vos propres métadonnées à chaque instantané sous forme de balises. Les balises vous permettent de classer vos instantanés de différentes manières, par exemple en fonction de leur objectif, de leur propriétaire ou de leur environnement. Cette approche est utile lorsque vous avez de nombreuses ressources de même type. Elle vous permet d'identifier rapidement une ressource spécifique en fonction des balises que vous lui avez attribuées. Pour plus d'informations, consultez Ressources que vous pouvez étiqueter.

Les étiquettes de répartition des coûts sont un moyen de suivre vos coûts sur plusieurs AWS services en regroupant vos dépenses sur les factures par valeur de balise. Pour en savoir plus sur les balises de répartition des coûts, veuillez consulter Utilisation des balises de répartition des coûts.

À l'aide de la console MemoryDB, de l'API MemoryDB AWS CLI, vous pouvez ajouter, répertorier, modifier, supprimer ou copier des balises de répartition des coûts sur vos instantanés. Pour plus d'informations, consultez Surveillance des coûts avec des balises de répartition des coûts.

Suppression d'un instantané

Un instantané automatique est automatiquement supprimé à l'expiration de sa limite de conservation. Si vous supprimez un cluster, tous ses instantanés automatiques sont également supprimés.

MemoryDB fournit une opération d'API de suppression qui vous permet de supprimer un instantané à tout moment, qu'il ait été créé automatiquement ou manuellement. Les instantanés manuels n'étant pas soumis à une limite de conservation, la suppression manuelle est le seul moyen de les supprimer.

Vous pouvez supprimer un instantané à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Supprimer un instantané (console)

La procédure suivante supprime un instantané à l'aide de la console MemoryDB.

Suppression d’un instantané

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Snapshots.

   L'écran Instantanés apparaît avec la liste de vos instantanés.
3. Cliquez sur le bouton radio situé à gauche du nom de l'instantané que vous souhaitez supprimer.
4. Choisissez Actions, puis Delete (Supprimer).
5. Si vous souhaitez supprimer cet instantané, entrez `delete` dans la zone de texte, puis choisissez Supprimer. Pour annuler la suppression, choisissez Annuler. L'état passe à deleting.

Supprimer un instantané (AWS CLI)

Utilisez l' AWS CLI opération delete-snapshot avec le paramètre suivant pour supprimer un instantané.

- `--snapshot-name`— Nom de l'instantané à supprimer.

Le code suivant supprime le `myBackup` cliché.

```
aws memorydb delete-snapshot --snapshot-name myBackup
```

Pour plus d'informations, veuillez consulter delete-snapshot dans la Référence des commandes AWS CLI .

Supprimer un instantané (API MemoryDB)

Utilisez l'opération `DeleteSnapshot` API avec le paramètre suivant pour supprimer un instantané.

- `SnapshotName`— Nom de l'instantané à supprimer.

Le code suivant supprime le `myBackup` cliché.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DeleteSnapshot
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SnapshotName=myBackup
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DeleteSnapshot.

Mise à l'échelle

La quantité de données dont votre application a besoin pour fonctionner est rarement statique. Elle augmente et diminue au fur et mesure du développement et des fluctuations normales liées à votre activité. Si vous gérez vous-même vos applications, vous devez fournir suffisamment de matériel pour faire face aux pics de demande, ce qui peut s'avérer coûteux. En utilisant MemoryDB, vous pouvez évoluer pour répondre à la demande actuelle, en ne payant que pour ce que vous utilisez.

Ce qui suit vous permet de trouver la rubrique appropriée pour les actions de mise à l'échelle que vous souhaitez exécuter.

Mise à l'échelle de MemoryDB

Action	Base de données de mémoire	
Augmentation d'échelle	Repartage en ligne et rééquilibrage des partitions pour MemoryDB	

Action	Base de données de mémoire	
Modification des types de nœuds	Dimensionnement vertical en ligne en modifiant le type de nœud	
Modification du nombre de partitions	Dimensionnement des clusters MemoryDB	

Dimensionnement des clusters MemoryDB

À mesure que la demande de vos clusters évolue, vous pouvez décider d'améliorer les performances ou de réduire les coûts en modifiant le nombre de partitions de votre cluster MemoryDB. Il est recommandé d'utiliser à cette fin la mise à l'échelle horizontal en ligne, parce que votre cluster peut ainsi continuer à traiter les demandes pendant le processus de mise à l'échelle.

Les conditions qui peuvent vous conduire à décider de redimensionner votre cluster sont les suivantes :

- Pression mémoire :

  Si les nœuds de votre cluster sont sous pression mémoire, vous pouvez décider de l'augmenter de telle sorte que vous ayez plus de ressources pour mieux stocker les données et traiter les demandes.

  Vous pouvez déterminer si vos nœuds sont soumis à une pression de mémoire en surveillant les indicateurs suivants : FreeableMemorySwapUsage,, et BytesUsedForMemoryDB.
- Goulet d'étranglement UC ou réseau :

  Si des problèmes de latence/débit affectent votre cluster, il se peut que vous ayez besoin de procéder à un agrandissement pour résoudre les problèmes.

  Vous pouvez surveiller vos niveaux de latence et de débit en surveillant les métriques suivantes : CPUUtilization,, NetworkBytesInNetworkBytesOut, CurrConnectionset. NewConnections
- Votre cluster est surdimensionné :

  La demande courante sur votre cluster est telle que la mise à l'échelle ne nuit pas aux performances et réduit vos coûts.

  Vous pouvez surveiller l'utilisation de votre cluster pour déterminer si vous pouvez ou non l'adapter en toute sécurité à l'aide des métriques suivantes : FreeableMemory, SwapUsage, BytesUsedForMemoryDB, CPUUtilization, NetworkBytesIn, NetworkBytesOutCurrConnections, et. NewConnections

Impact la mise à l'échelle sur les performances

Lorsque vous dimensionnez à l'aide du processus hors ligne, votre cluster se retrouve hors ligne pendant une partie importante du processus et de ce fait vous ne pouvez pas traiter les demandes.

Lorsque vous mettez à l'échelle à l'aide de la méthode en ligne, comme la mise à l'échelle est une opération gourmande en ressources de calcul, il en résulte une certaine dégradation des performances ; néanmoins, votre cluster continue à traiter les demandes d'un bout à l'autre de l'opération de mise à l'échelle. L'importance de la dégradation à laquelle vous êtes confronté dépend de votre utilisation normale de l'UC et de vos données.

Il existe deux manières de redimensionner votre cluster MemoryDB : la mise à l'échelle horizontale et la mise à l'échelle verticale.

- La mise à l'échelle horizontale vous permet de modifier le nombre de partitions dans le cluster en ajoutant ou en supprimant des partitions. Le processus de repartitionnement en ligne permet d'augmenter/de réduire le cluster pendant qu'il continue de répondre aux demandes entrantes.
- Dimensionnement vertical : modifier le type de nœud pour redimensionner le cluster. Le dimensionnement vertical en ligne permet d'augmenter/de réduire le cluster pendant qu'il continue de répondre aux demandes entrantes.

Si vous réduisez la taille et la capacité de mémoire du cluster, en augmentant ou en diminuant la taille, assurez-vous que la nouvelle configuration dispose de suffisamment de mémoire pour vos données et pour la surcharge de Redis OSS.

Repartage hors ligne et rééquilibrage des partitions pour MemoryDB

Le principal avantage de la reconfiguration de partitions hors ligne est que vous pouvez faire bien plus que simplement ajouter ou supprimer des partitions de votre cluster. Lorsque vous redéfinissez une partition hors ligne, vous pouvez non seulement modifier le nombre de partitions de votre cluster, mais également effectuer les opérations suivantes :

- Modifiez le type de nœud de votre cluster.
- Mettre à niveau vers une version plus récente du moteur.

> **Note**
>
> Le repartage hors ligne n'est pas pris en charge sur les clusters sur lesquels la hiérarchisation des données est activée. Pour plus d’informations, consultez Mise à niveau des données..

Le principal désavantage de la reconfiguration hors ligne des partitions est que votre cluster est hors ligne depuis la partie restauration du processus jusqu'à ce que vous mettiez à jour les points de terminaison de votre application. La durée pendant laquelle votre cluster est hors ligne varie avec la quantité de données de votre cluster.

Pour reconfigurer votre cluster MemoryDB hors ligne

1. Créez un instantané manuel de votre cluster MemoryDB existant. Pour plus d'informations, consultez Création d'instantanés manuels.
2. Créez un nouveau cluster en effectuant une restauration à partir de l'instantané. Pour plus d'informations, consultez Restaurer à partir d'un instantané.
3. Mettez à jour les points de terminaison dans votre application sur les points de terminaison du nouveau cluster. Pour plus d'informations, consultez Recherche de points de terminaison de connexion.

Repartage en ligne et rééquilibrage des partitions pour MemoryDB

En utilisant le repartage en ligne et le rééquilibrage des partitions avec MemoryDB, vous pouvez redimensionner votre MemoryDB de manière dynamique sans interruption de service. Cette approche signifie que votre cluster peut continuer à traiter des demandes même lorsqu'une mise à l'échelle ou un rééquilibrage est en cours.

Vous pouvez effectuer les actions suivantes :

- Extensibilité : augmentez la capacité de lecture et d'écriture en ajoutant des partitions à votre cluster MemoryDB.

  Si vous ajoutez une ou plusieurs partitions à votre cluster, le nombre de nœuds de chaque nouvelle partition est identique au nombre de nœuds de la plus petite partition existante.
- Mise à l'échelle : réduisez la capacité de lecture et d'écriture, et donc les coûts, en supprimant les partitions de votre cluster MemoryDB.

Actuellement, les restrictions suivantes s'appliquent au repartage en ligne de MemoryDB :

- Il existe des limitations pour les emplacements et les espaces de clés ou les éléments volumineux :

Si l'une des clés d'une partition contient un élément volumineux, cette clé ne peut pas faire l'objet d'une nouvelle migration lors d'une montée en charge ou d'un rééquilibrage. Cette fonctionnalité peut se traduire par des partitions non équilibrées.

Si l'une des clés d'une partition contient un élément volumineux (supérieur à 256 Mo après sérialisation), cette partition n'est pas supprimée lors de l'agrandissement. Cette fonctionnalité peut se traduire par le fait que certaines partitions ne sont pas supprimées.

- Lors de la mise à l'échelle, le nombre de nœuds dans les nouvelles partitions est égal au nombre de nœuds dans les partitions existantes.

Pour plus d'informations, consultez Bonnes pratiques : redimensionnement des clusters en ligne.

Vous pouvez redimensionner ou rééquilibrer horizontalement vos clusters MemoryDB à l'aide de l'API MemoryDB et de l' AWS CLI API MemoryDB. AWS Management Console

Ajout de partitions avec le repartitionnement en ligne

Vous pouvez ajouter des partitions à votre cluster MemoryDB à l'aide de l'API AWS Management Console AWS CLI, ou MemoryDB.

Ajout de partitions (console)

Vous pouvez utiliser le AWS Management Console pour ajouter une ou plusieurs partitions à votre cluster MemoryDB. La procédure suivante décrit le processus.

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans la liste des clusters, choisissez le nom du cluster à partir duquel vous souhaitez ajouter une partition.
3. Sous l'onglet Partitions et nœuds, choisissez Ajouter/Supprimer des partitions
4. Dans Nouveau nombre de partitions, entrez le nombre de partitions que vous souhaitez.
5. Choisissez Confirmer pour conserver les modifications ou Annuler pour les ignorer.

Ajout de partitions (AWS CLI)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster MemoryDB en ajoutant des partitions à l'aide du. AWS CLI

Utilisez les paramètres suivants avec `update-cluster`.

Paramètres

- `--cluster-name` : obligatoire. Spécifie le cluster (cluster) sur lequel l'opération de reconfiguration de partition doit être effectuée.
- `--shard-configuration` : obligatoire. Permet de définir le nombre de partitions.
  - `ShardCount`— Définissez cette propriété pour spécifier le nombre de partitions que vous souhaitez.

Example

L'exemple suivant modifie le nombre de partitions du cluster `my-cluster` à 2.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --shard-configuration \
        ShardCount=2
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --shard-configuration ^
        ShardCount=2
```

Elle renvoie la réponse JSON suivante :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
```

```
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Pour afficher les détails du cluster mis à jour une fois que son statut passe de mise à jour à disponible, utilisez la commande suivante :

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Il renverra la réponse JSON suivante :

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 2,
            "Shards": [
                {
                    "Name": "0001",
```

```
                    "Status": "available",
                    "Slots": "0-8191",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                },
                {
                    "Name": "0002",
                    "Status": "available",
                    "Slots": "8192-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0002-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-22T14:26:18.693000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
```

```
                        {
                            "Name": "my-cluster-0002-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-22T14:26:18.765000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Pour plus d'informations, consultez update-cluster dans la référence des AWS CLI commandes.

Ajouter des partitions (API MemoryDB)

Vous pouvez utiliser l'API MemoryDB pour reconfigurer les partitions de votre cluster MemoryDB en ligne à l'aide de l'opération. `UpdateCluster`

Utilisez les paramètres suivants avec `UpdateCluster`.

Paramètres

- `ClusterName` : obligatoire. Spécifie sur quel cluster l'opération de reconfiguration de partition doit être effectuée.
- `ShardConfiguration` : obligatoire. Permet de définir le nombre de partitions.
  - `ShardCount`— Définissez cette propriété pour spécifier le nombre de partitions que vous souhaitez.

Pour plus d'informations, consultez UpdateCluster.

Suppression de partitions avec le repartitionnement en ligne

Vous pouvez supprimer des partitions de votre cluster MemoryDB à l'aide de l'API AWS Management Console AWS CLI, ou MemoryDB.

Suppression de partitions (console)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster MemoryDB en supprimant les partitions à l'aide du. AWS Management Console

> **Important**
>
> Avant de supprimer des partitions de votre cluster, MemoryDB s'assure que toutes vos données rentreront dans les partitions restantes. Si les données sont correctes, les fragments sont supprimés du cluster comme demandé. Si les données ne rentrent pas dans les partitions restantes, le processus est interrompu et le cluster se retrouve avec la même configuration de partition qu'avant la demande.

Vous pouvez utiliser le AWS Management Console pour supprimer une ou plusieurs partitions de votre cluster MemoryDB. Vous ne pouvez pas supprimer tous les fragments d'un cluster. Vous devez plutôt supprimer le cluster. Pour plus d'informations, consultez Étape 4 : Supprimer un cluster. La procédure suivante décrit le processus de suppression d'une ou de plusieurs partitions.

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

2. Dans la liste des clusters, choisissez le nom du cluster dont vous souhaitez supprimer une partition.
3. Sous l'onglet Partitions et nœuds, choisissez Ajouter/Supprimer des partitions
4. Dans Nouveau nombre de partitions, entrez le nombre de partitions que vous souhaitez (avec un minimum de 1).
5. Choisissez Confirmer pour conserver les modifications ou Annuler pour les ignorer.

Suppression de partitions (AWS CLI)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster MemoryDB en supprimant les partitions à l'aide du. AWS CLI

> **Important**
>
> Avant de supprimer des partitions de votre cluster, MemoryDB s'assure que toutes vos données rentreront dans les partitions restantes. Si les données sont correctes, les partitions sont supprimées du cluster comme demandé et leurs espaces clés sont mappés dans les partitions restantes. Si les données ne rentrent pas dans les partitions restantes, le processus est interrompu et le cluster se retrouve avec la même configuration de partition qu'avant la demande.

Vous pouvez utiliser le AWS CLI pour supprimer une ou plusieurs partitions de votre cluster MemoryDB. Vous ne pouvez pas supprimer tous les fragments d'un cluster. Vous devez plutôt supprimer le cluster. Pour plus d'informations, consultez Étape 4 : Supprimer un cluster.

Utilisez les paramètres suivants avec `update-cluster`.

Paramètres

- `--cluster-name` : obligatoire. Spécifie le cluster (cluster) sur lequel l'opération de reconfiguration de partition doit être effectuée.
- `--shard-configuration` : obligatoire. Permet de définir le nombre de partitions à l'aide de la `ShardCount` propriété :

  `ShardCount`— Définissez cette propriété pour spécifier le nombre de partitions que vous souhaitez.

Example

L'exemple suivant modifie le nombre de partitions du cluster `my-cluster` à 2.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --shard-configuration \
        ShardCount=2
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --shard-configuration ^
        ShardCount=2
```

Elle renvoie la réponse JSON suivante :

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
```

```
    }
}
```

Pour afficher les détails du cluster mis à jour une fois que son statut passe de mise à jour à disponible, utilisez la commande suivante :

Pour Linux, macOS ou Unix :

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Pour Windows :

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Il renverra la réponse JSON suivante :

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 2,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-8191",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
```

```
                        }
                    },
                    {
                        "Name": "my-cluster-0001-002",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1b",
                        "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    }
                ],
                "NumberOfNodes": 2
            },
            {
                "Name": "0002",
                "Status": "available",
                "Slots": "8192-16383",
                "Nodes": [
                    {
                        "Name": "my-cluster-0002-001",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1b",
                        "CreateTime": "2021-08-22T14:26:18.693000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    },
                    {
                        "Name": "my-cluster-0002-002",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1a",
                        "CreateTime": "2021-08-22T14:26:18.765000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    }
                ],
```

```
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Pour plus d'informations, consultez update-cluster dans la référence des AWS CLI commandes.

Supprimer des partitions (API MemoryDB)

Vous pouvez utiliser l'API MemoryDB pour reconfigurer les partitions de votre cluster MemoryDB en ligne à l'aide de l'opération. `UpdateCluster`

Le processus suivant décrit comment reconfigurer les partitions de votre cluster MemoryDB en supprimant les partitions à l'aide de l'API MemoryDB.

> **Important**
>
> Avant de supprimer des partitions de votre cluster, MemoryDB s'assure que toutes vos données rentreront dans les partitions restantes. Si les données sont correctes, les partitions sont supprimées du cluster comme demandé et leurs espaces clés sont mappés dans les partitions restantes. Si les données ne rentrent pas dans les partitions restantes, le processus

est interrompu et le cluster se retrouve avec la même configuration de partition qu'avant la demande.

Vous pouvez utiliser l'API MemoryDB pour supprimer une ou plusieurs partitions de votre cluster MemoryDB. Vous ne pouvez pas supprimer tous les fragments d'un cluster. Vous devez plutôt supprimer le cluster. Pour plus d'informations, consultez Étape 4 : Supprimer un cluster.

Utilisez les paramètres suivants avec `UpdateCluster`.

Paramètres

- `ClusterName` : obligatoire. Spécifie le cluster (cluster) sur lequel l'opération de reconfiguration de partition doit être effectuée.
- `ShardConfiguration` : obligatoire. Permet de définir le nombre de partitions à l'aide de la `ShardCount` propriété :

  `ShardCount`— Définissez cette propriété pour spécifier le nombre de partitions que vous souhaitez.

Dimensionnement vertical en ligne en modifiant le type de nœud

En utilisant la mise à l'échelle verticale en ligne avec MemoryDB, vous pouvez faire évoluer votre cluster de manière dynamique avec un temps d'arrêt minimal. Cela permet à votre cluster de répondre aux demandes même lors de la mise à l'échelle.

> **Note**
>
> La mise à l'échelle n'est pas prise en charge entre un cluster de hiérarchisation des données (par exemple, un cluster utilisant un type de nœud r6gd) et un cluster qui n'utilise pas la hiérarchisation des données (par exemple, un cluster utilisant un type de nœud r6g). Pour plus d'informations, consultez Mise à niveau des données.

Vous pouvez effectuer les actions suivantes :

- Augmenter la capacité de lecture et d'écriture : augmentez la capacité de lecture et d'écriture en ajustant le type de nœud de votre cluster MemoryDB pour utiliser un type de nœud plus important.

MemoryDB redimensionne dynamiquement votre cluster tout en restant en ligne et en répondant aux demandes.

- Réduire : réduisez la capacité de lecture et d'écriture en ajustant le type de nœud pour votre cluster Redis afin d'utiliser un type de nœud plus petit. Encore une fois, MemoryDB redimensionne dynamiquement votre cluster tout en restant en ligne et en répondant aux demandes. Dans ce cas, vous réduisez vos coûts en diminuant la taille du nœud.

> **Note**
>
> Les processus d'augmentation et de réduction reposent sur la création de clusters avec des types de nœuds nouvellement sélectionnés et la synchronisation des nouveaux nœuds avec les anciens. Afin de garantir un flux d'augmentation/de réduction fluide, procédez comme suit :
>
> - Bien que le processus de dimensionnement vertical soit conçu pour rester entièrement en ligne, il repose sur la synchronisation des données entre l'ancien nœud et le nouveau. Nous vous recommandons d'initier l'augmentation/la réduction lorsqu'un trafic minimum des données est prévu.
> - Testez le comportement de votre application lors du repartitionnement dans un environnement intermédiaire, si possible.

Augmentation en ligne

Rubriques

- Mise à l'échelle des clusters MemoryDB (console)
- Mise à l'échelle des clusters MemoryDB (CLI)AWS
- Mise à l'échelle des clusters MemoryDB (API MemoryDB)

Mise à l'échelle des clusters MemoryDB (console)

La procédure suivante décrit comment augmenter la taille d'un cluster MemoryDB à l'aide du. AWS Management Console Au cours de ce processus, votre cluster MemoryDB continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour agrandir un cluster (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Choisissez le cluster dans la liste.
3. Choisissez Actions, puis Modifier.
4. Dans la boîte de dialogue Modifier le cluster :
   - Choisissez le type de nœud auquel vous souhaitez passer dans la liste Type de nœud. Pour l'augmenter, sélectionnez un type de nœud plus grand que votre nœud existant.
5. Sélectionnez Enregistrer les modifications.

   Le statut du cluster passe à celui de modification. Lorsque son statut passe à available, la modification est terminée et vous pouvez commencer à utiliser le nouveau cluster.

Mise à l'échelle des clusters MemoryDB (CLI)AWS

La procédure suivante décrit comment augmenter la taille d'un cluster MemoryDB à l'aide du. AWS CLI Au cours de ce processus, votre cluster MemoryDB continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour augmenter la taille d'un cluster MemoryDB (CLI)AWS

1. Déterminez les types de nœuds que vous pouvez augmenter en exécutant la AWS CLI `list-allowed-node-type-updates` commande avec le paramètre suivant.

   Pour Linux, macOS ou Unix :

   ```
   aws memorydb list-allowed-node-type-updates \
        --cluster-name my-cluster-name
   ```

   Pour Windows :

   ```
   aws memorydb list-allowed-node-type-updates ^
        --cluster-name my-cluster-name
   ```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
     "ScaleUpNodeTypes": [
         "db.r6g.2xlarge",
         "db.r6g.large"
     ],
     "ScaleDownNodeTypes": [
         "db.r6g.large"
     ],
}
```

Pour plus d'informations, consultez list-allowed-node-type-updates dans la AWS CLI référence.

2. Modifiez votre cluster pour l'adapter au nouveau type de nœud plus grand à l'aide de la AWS CLI `update-cluster` commande et des paramètres suivants.

   - `--cluster-name`— Le nom du cluster vers lequel vous procédez à la mise à l'échelle.
   - `--node-type`— Le nouveau type de nœud pour lequel vous souhaitez redimensionner le cluster. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-updates` lors de l'étape 1.

   Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster  \
 --cluster-name my-cluster \
 --node-type db.r6g.2xlarge
```

   Pour Windows :

```
aws memorydb update-cluster ^
     --cluster-name my-cluster ^
     --node-type db.r6g.2xlarge ^
```

   Pour plus d'informations, consultez update-cluster.

Mise à l'échelle des clusters MemoryDB (API MemoryDB)

Le processus suivant fait passer votre cluster de son type de nœud actuel à un nouveau type de nœud plus grand à l'aide de l'API MemoryDB. Au cours de ce processus, MemoryDB met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster continue de rester en ligne et de répondre aux demandes entrantes.

Le temps nécessaire pour passer à un type de nœud plus important varie en fonction de votre type de nœud et de la quantité de données dans votre cluster actuel.

Pour augmenter la taille d'un cluster MemoryDB (API MemoryDB)

1. Déterminez les types de nœuds que vous pouvez augmenter à l'aide de l'`ListAllowedNodeTypeUpdates`action d'API MemoryDB avec le paramètre suivant.

   - `ClusterName`— le nom du cluster. Utilisez ce paramètre pour décrire un cluster spécifique plutôt que tous les clusters.

   ```
   https://memory-db.us-east-1.amazonaws.com/
       ?Action=ListAllowedNodeTypeUpdates
       &ClusterName=MyCluster
       &Version=2021-01-01
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20210802T192317Z
       &X-Amz-Credential=<credential>
   ```

   Pour plus d'informations, consultez la ListAllowedNodeTypeUpdatesréférence de l'API MemoryDB.

2. Faites évoluer votre cluster actuel jusqu'au nouveau type de nœud à l'aide de l'action de l'API `UpdateCluster` MemoryDB et avec les paramètres suivants.

   - `ClusterName`— le nom du cluster.
   - `NodeType`— le nouveau type de nœud plus grand des clusters de ce cluster. Cette valeur doit correspondre à l'un des types d'instance renvoyés par l'action `ListAllowedNodeTypeUpdates` lors de l'étape 1.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &NodeType=db.r6g.2xlarge
    &ClusterName=myCluster
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez UpdateCluster.

Réduction en ligne

Rubriques

- Réduction de la taille des clusters MemoryDB (console)
- Réduction de la taille des clusters MemoryDB (CLI)AWS
- Réduction de la taille des clusters MemoryDB (API MemoryDB)

Réduction de la taille des clusters MemoryDB (console)

La procédure suivante décrit comment réduire la taille d'un cluster MemoryDB à l'aide du. AWS Management Console Au cours de ce processus, votre cluster MemoryDB continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour réduire la taille d'un cluster MemoryDB (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Choisissez votre cluster préféré dans la liste.
3. Choisissez Actions, puis Modifier.

4. Dans la boîte de dialogue Modifier le cluster :

   - Choisissez le type de nœud auquel vous souhaitez passer dans la liste Type de nœud. Pour le réduire, sélectionnez un type de nœud plus petit que votre nœud existant. Notez que tous les types de nœuds ne sont pas disponibles pour la réduction de la capacité.

5. Sélectionnez Enregistrer les modifications.

   Le statut du cluster passe à celui de modification. Lorsque son statut passe à available, la modification est terminée et vous pouvez commencer à utiliser le nouveau cluster.

Réduction de la taille des clusters MemoryDB (CLI)AWS

La procédure suivante décrit comment réduire la taille d'un cluster MemoryDB à l'aide du. AWS CLI Au cours de ce processus, votre cluster MemoryDB continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour réduire la taille d'un cluster MemoryDB (CLI)AWS

1. Déterminez les types de nœuds que vous pouvez réduire en exécutant la AWS CLI `list-allowed-node-type-updates` commande avec le paramètre suivant.

   Pour Linux, macOS ou Unix :

   ```
   aws memorydb list-allowed-node-type-updates \
        --cluster-name my-cluster-name
   ```

   Pour Windows :

   ```
   aws memorydb list-allowed-node-type-updates ^
        --cluster-name my-cluster-name
   ```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

   ```
   {
        "ScaleUpNodeTypes": [
            "db.r6g.2xlarge",
            "db.r6g.large"
        ],
        "ScaleDownNodeTypes": [
            "db.r6g.large"
   ```

```
    ],
}
```

Pour plus d'informations, consultez list-allowed-node-type-updates.

2. Modifiez votre cluster pour le réduire au nouveau type de nœud plus petit, à l'aide de la `update-cluster` commande et des paramètres suivants.

   - `--cluster-name`— Le nom du cluster vers lequel vous réduisez la taille.
   - `--node-type`— Le nouveau type de nœud pour lequel vous souhaitez redimensionner le cluster. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-updates` lors de l'étape 1.

   Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster  \
     --cluster-name my-cluster \
     --node-type db.r6g.large
```

   Pour Windows :

```
aws memorydb update-cluster ^
     --cluster-name my-cluster ^
     --node-type db.r6g.large
```

   Pour plus d'informations, consultez update-cluster.

Réduction de la taille des clusters MemoryDB (API MemoryDB)

Le processus suivant fait passer votre cluster de son type de nœud actuel à un nouveau type de nœud plus petit à l'aide de l'API MemoryDB. Au cours de ce processus, votre cluster MemoryDB continuera à traiter les demandes avec un temps d'arrêt minimal.

Le temps nécessaire pour passer à un type de nœud plus petit varie en fonction de votre type de nœud et de la quantité de données dans votre cluster actuel.

Réduction de la taille (API MemoryDB)

1. Déterminez les types de nœuds que vous pouvez réduire à l'aide de l'ListAllowedNodeTypeUpdatesAPI avec le paramètre suivant :

   - `ClusterName`— le nom du cluster. Utilisez ce paramètre pour décrire un cluster spécifique plutôt que tous les clusters.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=ListAllowedNodeTypeUpdates
    &ClusterName=MyCluster
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T192317Z
    &X-Amz-Credential=<credential>
```

2. Réduisez votre cluster actuel au nouveau type de nœud à l'aide de l'UpdateClusterAPI avec les paramètres suivants.

   - `ClusterName`— le nom du cluster.
   - `NodeType`— le nouveau type de nœud plus petit des clusters de ce cluster. Cette valeur doit correspondre à l'un des types d'instance renvoyés par l'action `ListAllowedNodeTypeUpdates` lors de l'étape 1.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &NodeType=db.r6g.2xlarge
    &ClusterName=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Configuration des paramètres de moteur à l'aide de groupes de paramètres

MemoryDB utilise des paramètres pour contrôler les propriétés d'exécution de vos nœuds et clusters. Habituellement, les dernières versions de moteurs comprennent des paramètres supplémentaires pour prendre en charge une fonctionnalité plus récente. Pour plus de détails sur les tableaux de paramètres, consultez Paramètres spécifiques à Redis OSS.

Bien entendu, certaines valeurs de paramètres telles que `maxmemory` sont déterminées par le type de nœud de et de moteur. Pour un tableau des valeurs de ces paramètres de type de nœud, consultez Paramètres spécifiques au type de nœud MemoryDB.

Rubriques

- Gestion des paramètres
- Niveaux de groupe de paramètres
- Création d'un groupe de paramètres
- Liste des groupes de paramètres par nom
- Affichage des valeurs d'un groupe de paramètres
- Modification d'un groupe de paramètres
- Suppression d'un groupe de paramètres
- Paramètres spécifiques à Redis OSS

Gestion des paramètres

Les paramètres sont regroupés dans les groupes de paramètre nommés pour faciliter la gestion paramètre. Un groupe de paramètres représente une combinaison de valeurs spécifiques pour les paramètres qui sont transmis au logiciel de moteur de au moment du démarrage. Ces valeurs déterminent le comportement des processus du moteur sur chaque nœud au moment de l'exécution. Les valeurs des paramètres sur un groupe de paramètres spécifiques s'appliquent à tous les nœuds associés au groupe, indépendamment du cluster auquel ils appartiennent.

Pour affiner les performances de votre cluster, vous pouvez modifier certaines valeurs des paramètres ou modifier le groupe de paramètres du cluster.

- Vous ne pouvez pas modifier, ni supprimer les groupes de paramètres par défaut. Si vous avez besoin de valeurs des paramètres personnalisés, vous devez créer un groupe de paramètres personnalisés.
- La famille de groupe de paramètres et le cluster que vous lui associez doivent être compatibles. Par exemple, si votre cluster exécute Redis OSS version 6, vous ne pouvez utiliser que des groupes de paramètres, par défaut ou personnalisés, de la famille memorydb_redis6.
- Lorsque vous modifiez les paramètres d'un cluster, la modification est immédiatement appliquée au cluster. C'est vrai si vous changez le groupe de paramètres même du cluster ou une valeur de paramètre dans le groupe de paramètres du cluster.

Niveaux de groupe de paramètres

Niveaux du groupe de paramètres MemoryDB

Par défaut global

Le groupe de paramètres racine de premier niveau pour tous les clients MemoryDB de la région.

Le groupe de paramètres global par défaut :

- Est réservé à MemoryDB et n'est pas disponible pour le client.

Par défaut client

Une copie du groupe de paramètres par défaut global créé pour l'usage du client.

Le groupe de paramètres par défaut du client :

- Est créé et détenu par MemoryDB.
- Le client peut l'utiliser en tant que groupe de paramètres pour tous les clusters exécutant une version de moteur prise en charge par ce groupe de paramètres.
- Ne peut pas être modifié par le client.

Appartient au client

Une copie du groupe de paramètres Customer Default. Un groupe de paramètres appartenant au client est créé chaque fois que le client crée un groupe de paramètres.

Le groupe de paramètres appartenant au client :

- Est créé par le client et lui appartient.
- Peut être affecté à tout cluster compatible du client.
- Peut être modifié par le client pour créer un groupe de paramètres personnalisé.

  Toutes les valeurs de paramètre ne peuvent pas être modifiées. Pour plus d’informations, consultez Paramètres spécifiques à Redis OSS.

Création d'un groupe de paramètres

Vous devez créer un groupe de paramètres s'il existe une ou plusieurs valeurs de paramètre que vous voulez changer par rapport aux valeurs par défaut. Vous pouvez créer un groupe de paramètres à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Création d'un groupe de paramètres (console)

La procédure suivante montre comment créer un groupe de paramètres à l'aide de la console MemoryDB.

Pour créer un groupe de paramètres à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Groupes de paramètres pour consulter la liste des groupes de paramètres disponibles.
3. Pour créer un groupe de paramètres, choisissez Créer un groupe de paramètres.

   La page Créer un groupe de paramètres apparaît.
4. Dans la zone Name, tapez un nom unique pour ce groupe de paramètres.

   Lors de création d'un cluster ou de la modification d'un groupe de paramètres d'un cluster, vous choisissez le groupe de paramètres par son nom. Par conséquent, nous recommandons que le nom soit informatif et permette d'identifier la famille du groupe de paramètres.

   Contraintes d'attribution de nom à un groupe de paramètres :

   - Doit commencer par une lettre ASCII.
   - Elle ne peut contenir que des lettres ASCII, des chiffres et des tirets ('-').
   - Doit être comprise entre 1 et 255 caractères.
   - Ils ne peuvent pas comporter deux traits d'union consécutifs.
   - Ils ne peuvent pas se terminer par un trait d'union.
5. Dans la zone Description, saisissez une description du groupe de paramètres.
6. Dans la zone de compatibilité des versions Redis OSS, choisissez une version du moteur à laquelle correspond ce groupe de paramètres.

7. Dans les balises, ajoutez éventuellement des balises pour rechercher et filtrer vos groupes de paramètres ou suivre vos AWS coûts.
8. Choisissez Créer pour créer le groupe de paramètres.

   Pour terminer le processus sans créer le groupe de paramètres, choisissez Annuler.
9. Lorsque le groupe de paramètres est créé, il a les valeurs par défaut de la famille. Pour modifier les valeurs par défaut, vous devez modifier le groupe de paramètres. Pour plus d'informations, consultez Modification d'un groupe de paramètres.

Création d'un groupe de paramètres (AWS CLI)

Pour créer un groupe de paramètres à l'aide de AWS CLI, utilisez la commande `create-parameter-group` avec ces paramètres.

- `--parameter-group-name` – Le nom du groupe de paramètres.

  Contraintes d'attribution de nom à un groupe de paramètres :
  - Doit commencer par une lettre ASCII.
  - Elle ne peut contenir que des lettres ASCII, des chiffres et des tirets ('-').
  - Doit être comprise entre 1 et 255 caractères.
  - Ils ne peuvent pas comporter deux traits d'union consécutifs.
  - Ils ne peuvent pas se terminer par un trait d'union.
- `--family` – La famille du moteur et de version pour le groupe de paramètres.
- `--description` – Une description fourni par l'utilisateur pour le groupe de paramètres.

Example

L'exemple suivant crée un groupe de paramètres nommé MyRedis6x en utilisant la famille memorydb_redis6 comme modèle.

Pour Linux, macOS ou Unix :

```
aws memorydb create-parameter-group \
    --parameter-group-name myRedis6x  \
    --family memorydb_redis6 \
    --description "My first parameter group"
```

Pour Windows :

```
aws memorydb create-parameter-group ^
    --parameter-group-name myRedis6x  ^
    --family memorydb_redis6 ^
    --description "My first parameter group"
```

Le résultat de cette commande devrait ressembler à cet exemple.

```
{
    "ParameterGroup": {
        "Name": "myRedis6x",
        "Family": "memorydb_redis6",
        "Description": "My first parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x"
    }
}
```

Lorsque le groupe de paramètres est créé, il a les valeurs par défaut de la famille. Pour modifier les valeurs par défaut, vous devez modifier le groupe de paramètres. Pour plus d'informations, consultez Modification d'un groupe de paramètres.

Pour plus d'informations, consultez `create-parameter-group`.

Création d'un groupe de paramètres (API MemoryDB)

Pour créer un groupe de paramètres à l'aide de l'API MemoryDB, utilisez l'`CreateParameterGroup`action avec ces paramètres.

- `ParameterGroupName` – Le nom du groupe de paramètres.

  Contraintes d'attribution de nom à un groupe de paramètres :

  - Doit commencer par une lettre ASCII.
  - Elle ne peut contenir que des lettres ASCII, des chiffres et des tirets ('-').
  - Doit être comprise entre 1 et 255 caractères.
  - Ils ne peuvent pas comporter deux traits d'union consécutifs.
  - Ils ne peuvent pas se terminer par un trait d'union.
- `Family` – La famille du moteur et de version pour le groupe de paramètres. Par exemple, `memorydb_redis6`.

- `Description` – Une description fourni par l'utilisateur pour le groupe de paramètres.

Example

L'exemple suivant crée un groupe de paramètres nommé MyRedis6x en utilisant la famille memorydb_redis6 comme modèle.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=CreateParameterGroup
   &Family=memorydb_redis6
   &ParameterGroupName=myRedis6x
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action devrait se présenter comme suit.

```
<CreateParameterGroupResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <CreateParameterGroupResult>
    <ParameterGroup>
      <Name>myRedis6x</Name>
      <Family>memorydb_redis6</Family>
      <Description>My first parameter group</Description>
      <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
    </ParameterGroup>
  </CreateParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateParameterGroupResponse>
```

Lorsque le groupe de paramètres est créé, il a les valeurs par défaut de la famille. Pour modifier les valeurs par défaut, vous devez modifier le groupe de paramètres. Pour plus d’informations, consultez Modification d'un groupe de paramètres.

Pour plus d’informations, consultez `CreateParameterGroup`.

Liste des groupes de paramètres par nom

Vous pouvez répertorier les groupes de paramètres à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Liste des groupes de paramètres par nom (console)

La procédure suivante montre comment afficher la liste des groupes de paramètres à l'aide de la console MemoryDB.

Pour répertorier les groupes de paramètres à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Groupes de paramètres pour consulter la liste des groupes de paramètres disponibles.

Répertorier les groupes de paramètres par nom (AWS CLI)

Pour générer une liste de groupes de paramètres à l'aide de AWS CLI, utilisez la commande`describe-parameter-groups`. Si vous fournissez le nom d'un groupe de paramètres, seul ce groupe de paramètres sera répertorié. Si vous ne fournissez pas de nom d'un groupe de paramètres, un maximum de `--max-results` groupes de paramètres sera répertorié. Dans les deux cas, le nom, la famille et la description du groupe de paramètres sont répertoriés.

Example

L'exemple de code suivant répertorie le groupe de paramètres MyRedis6x.

Pour Linux, macOS ou Unix :

```
aws memorydb describe-parameter-groups \
    --parameter-group-name myRedis6x
```

Pour Windows :

```
aws memorydb describe-parameter-groups ^
    --parameter-group-name myRedis6x
```

Le résultat de cette commande se présentera de la façon suivante, avec le nom, la famille et la description du groupe de paramètres.

```
{
    "ParameterGroups": [
     {
         "Name": "myRedis6x",
         "Family": "memorydb_redis6",
         "Description": "My first parameter group",
                "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
     }
    ]
}
```

Example

L'exemple de code suivant répertorie le groupe de paramètres MyRedis6x pour les groupes de paramètres exécutés sur le moteur Redis OSS version 5.0.6 et ultérieure.

Pour Linux, macOS ou Unix :

```
aws memorydb describe-parameter-groups \
    --parameter-group-name myRedis6x
```

Pour Windows :

```
aws memorydb describe-parameter-groups ^
    --parameter-group-name myRedis6x
```

La sortie de cette commande ressemblera à ceci, répertoriant le nom, la famille et la description du groupe de paramètres.

```
{
    "ParameterGroups": [
     {
         "Name": "myRedis6x",
         "Family": "memorydb_redis6",
         "Description": "My first parameter group",
                "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
```

```
     }
    ]
}
```

Example

L'exemple de code suivant répertorie jusqu'à 20 groupes de paramètres.

```
aws memorydb describe-parameter-groups --max-results 20
```

La sortie JSON de cette commande ressemblera à ceci, répertoriant le nom, la famille et la description de chaque groupe de paramètres.

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.memorydb-redis6",
            "Family": "memorydb_redis6",
            "Description": "Default parameter group for memorydb_redis6",
            "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
default.memorydb-redis6"
        },
        ...
    ]
}
```

Pour plus d'informations, consultez [describe-parameter-groups](#).

Répertorier les groupes de paramètres par nom (API MemoryDB)

Pour générer une liste de groupes de paramètres à l'aide de l'API MemoryDB, utilisez l'`DescribeParameterGroups`action. Si vous fournissez le nom d'un groupe de paramètres, seul ce groupe de paramètres sera répertorié. Si vous ne fournissez pas de nom d'un groupe de paramètres, un maximum de `MaxResults` groupes de paramètres sera répertorié. Dans les deux cas, le nom, la famille et la description du groupe de paramètres sont répertoriés.

Example

L'exemple de code suivant répertorie jusqu'à 20 groupes de paramètres.

```
https://memory-db.us-east-1.amazonaws.com/
```

```
   ?Action=DescribeParameterGroups
   &MaxResults=20
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La réponse de cette action ressemblera à ceci, indiquant le nom, la famille et la description dans le cas de memorydb_redis6, pour chaque groupe de paramètres.

```
<DescribeParameterGroupsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <DescribeParameterGroupsResult>
    <ParameterGroups>
      <ParameterGroup>
        <Name>myRedis6x</Name>
        <Family>memorydb_redis6</Family>
        <Description>My custom Redis OSS 6 parameter group</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
      </ParameterGroup>
       <ParameterGroup>
        <Name>default.memorydb-redis6</Name>
        <Family>memorydb_redis6</Family>
        <Description>Default parameter group for memorydb_redis6</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/default.memorydb-
redis6</ARN>
      </ParameterGroup>
    </ParameterGroups>
  </DescribeParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeParameterGroupsResponse>
```

Example

L'exemple de code suivant répertorie le groupe de paramètres MyRedis6x.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeParameterGroups
   &ParameterGroupName=myRedis6x
   &SignatureVersion=4
```

```
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La réponse à cette action se présentera de la façon suivante, avec le nom, la famille et la description.

```
<DescribeParameterGroupsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <DescribeParameterGroupsResult>
    <ParameterGroups>
      <ParameterGroup>
        <Name>myRedis6x</Name>
        <Family>memorydb_redis6</Family>
        <Description>My custom Redis OSS 6 parameter group</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
      </ParameterGroup>
    </ParameterGroups>
  </DescribeParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeParameterGroupsResponse>
```

Pour plus d'informations, consultez DescribeParameterGroups.

Affichage des valeurs d'un groupe de paramètres

Vous pouvez répertorier les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Affichage des valeurs d'un groupe de paramètres (console)

La procédure suivante montre comment répertorier les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la console MemoryDB.

Pour répertorier les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Groupes de paramètres pour consulter la liste des groupes de paramètres disponibles.
3. Choisissez le groupe de paramètres pour lequel vous souhaitez répertorier les paramètres et les valeurs en choisissant le nom (et non la case à côté) du nom du groupe de paramètres.

   Les paramètres et leurs valeurs figureront au bas de l'écran. En raison du nombre de paramètres, vous devrez peut-être faire défiler la liste vers le haut et en bas pour trouver le paramètre souhaité.

Lister les valeurs d'un groupe de paramètres (AWS CLI)

Pour répertorier les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de AWS CLI, utilisez la commande`describe-parameters`.

Example

L'exemple de code suivant répertorie tous les paramètres et leurs valeurs pour le groupe de paramètres MyRedis6x.

Pour Linux, macOS ou Unix :

```
aws memorydb describe-parameters \
    --parameter-group-name myRedis6x
```

Pour Windows :

```
aws memorydb describe-parameters ^
    --parameter-group-name myRedis6x
```

Pour plus d'informations, consultez `describe-parameters`.

Lister les valeurs d'un groupe de paramètres (API MemoryDB)

Pour répertorier les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de l'API MemoryDB, utilisez l'`DescribeParameters`action.

Pour plus d'informations, consultez `DescribeParameters`.

Modification d'un groupe de paramètres

> **Important**
>
> Vous ne pouvez pas modifier un groupe de paramètres par défaut.

Vous pouvez modifier certaines valeurs des paramètres dans un groupe de paramètres. Ces valeurs de ces paramètres sont appliquées aux clusters associés au groupe de paramètres. Pour savoir quand une modification de valeur de paramètre est appliquée à un groupe de paramètres, consultez Paramètres spécifiques à Redis OSS.

Modification d'un groupe de paramètres

La procédure suivante montre comment modifier la valeur du paramètre à l'aide de la console MemoryDB. Vous devez utiliser la même procédure pour modifier la valeur de tout paramètre.

Pour modifier la valeur d'un paramètre à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Groupes de paramètres pour consulter la liste des groupes de paramètres disponibles.
3. Choisissez le groupe de paramètres que vous souhaitez modifier en cliquant sur le bouton radio situé à gauche du nom du groupe de paramètres.

Choisissez Actions, puis Afficher les détails. Vous pouvez également choisir le nom du groupe de paramètres pour accéder à la page de détails.

4. Pour modifier le paramètre, choisissez Modifier. Tous les paramètres modifiables seront activés pour être édités. Vous devrez peut-être parcourir les pages pour trouver le paramètre que vous souhaitez modifier. Vous pouvez également rechercher le paramètre par son nom, sa valeur ou son type dans le champ de recherche.
5. Apportez les modifications de paramètres nécessaires.
6. Choisissez Enregistrer pour enregistrer les modifications.
7. Si vous avez modifié les valeurs des paramètres sur plusieurs pages, vous pouvez passer en revue toutes les modifications en choisissant Aperçu des modifications. Pour confirmer les modifications, choisissez Enregistrer les modifications. Pour apporter d'autres modifications, choisissez Retour.
8. La page de détails des paramètres vous permet également de rétablir les valeurs par défaut. Pour rétablir les valeurs par défaut, choisissez Rétablir les valeurs par défaut. Des cases à cocher apparaîtront sur le côté gauche de tous les paramètres. Vous pouvez sélectionner ceux que vous souhaitez réinitialiser et choisir Procéder à la réinitialisation pour confirmer.

   Choisissez Confirmer pour confirmer l'action de réinitialisation dans la boîte de dialogue.
9. La page de détails des paramètres vous permet de définir le nombre de paramètres que vous souhaitez voir sur chaque page. Utilisez la roue dentée sur le côté droit pour effectuer ces modifications. Vous pouvez également activer/désactiver les colonnes de votre choix sur la page de détails. Ces modifications sont maintenues pendant toute la durée de la session de la console.

   Pour rechercher le nom du paramètre que vous avez modifié, consultez Paramètres spécifiques à Redis OSS.

Modification d'un groupe de paramètres (AWS CLI)

Pour modifier la valeur d'un paramètre à l'aide de AWS CLI, utilisez la commande`update-parameter-group`.

Pour rechercher le nom du paramètre que vous avez modifié, ainsi que les valeurs autorisées, consultez Paramètres spécifiques à Redis OSS

Pour plus d'informations, consultez update-parameter-group.

Modification d'un groupe de paramètres (API MemoryDB)

Pour modifier les valeurs des paramètres d'un groupe de paramètres à l'aide de l'API MemoryDB, utilisez l'`UpdateParameterGroup`action.

Pour rechercher le nom du paramètre que vous avez modifié, ainsi que les valeurs autorisées, consultez Paramètres spécifiques à Redis OSS

Pour plus d’informations, consultez `UpdateParameterGroup`.

Suppression d'un groupe de paramètres

Vous pouvez supprimer un groupe de paramètres personnalisé à l'aide de la console MemoryDB, de l'API MemoryDB ou de l' AWS CLI API MemoryDB.

Vous ne pouvez pas supprimer un groupe de paramètres s'il est associé à n'importe quel clusters de Vous ne pouvez pas supprimer non plus les groupes de paramètres par défaut.

Suppression d'un groupe de paramètres (console)

La procédure suivante montre comment supprimer un groupe de paramètres à l'aide de la console MemoryDB.

Pour supprimer un groupe de paramètres à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Groupes de paramètres pour consulter la liste des groupes de paramètres disponibles.
3. Choisissez les groupes de paramètres que vous souhaitez supprimer en cliquant sur le bouton radio situé à gauche du nom du groupe de paramètres.

   Choisissez Actions, puis Delete (Supprimer).
4. L'écran de confirmation Delete Parameter Groups s'affichera.
5. Pour supprimer les groupes de paramètres, entrez Supprimer dans la zone de texte de confirmation.

   Pour conserver les groupes de paramètres, choisissez Annuler.

Supprimer un groupe de paramètres (AWS CLI)

Pour supprimer un groupe de paramètres à l'aide de AWS CLI, utilisez la commande`delete-parameter-group`. Pour le groupe de paramètres à supprimer, le groupe de paramètres spécifié par `--parameter-group-name` ne peut pas avoir de clusters associés, et ne peut pas être non plus un groupe de paramètres par défaut.

L'exemple de code suivant supprime le groupe de paramètres MyRedis6x.

Example

Pour Linux, macOS ou Unix :

```
aws memorydb delete-parameter-group \
    --parameter-group-name myRedis6x
```

Pour Windows :

```
aws memorydb delete-parameter-group ^
    --parameter-group-name myRedis6x
```

Pour plus d'informations, consultez delete-parameter-group.

Supprimer un groupe de paramètres (API MemoryDB)

Pour supprimer un groupe de paramètres à l'aide de l'API MemoryDB, utilisez l'`DeleteParameterGroup`action. Pour le groupe de paramètres à supprimer, le groupe de paramètres spécifié par `ParameterGroupName` ne peut pas avoir de clusters associés, et ne peut pas être non plus un groupe de paramètres par défaut.

Example

L'exemple de code suivant supprime le groupe de paramètres MyRedis6x.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DeleteParameterGroup
   &ParameterGroupName=myRedis6x
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Pour plus d’informations, consultez `DeleteParameterGroup`.

Paramètres spécifiques à Redis OSS

Si vous ne spécifiez pas de groupe de paramètres pour votre cluster Redis OSS, un groupe de paramètres par défaut adapté à la version de votre moteur sera utilisé. Vous ne pouvez pas modifier les valeurs des paramètres dans le groupe de paramètres par défaut. Vous pouvez cependant créer un groupe de paramètres personnalisés et l'assigner à votre cluster à tout moment, tant que les valeurs des paramètres modifiables sous conditions sont les mêmes dans les deux groupes de paramètres. Pour plus d'informations, consultez Création d'un groupe de paramètres.

Rubriques

- Modifications des paramètres de Redis OSS 7
- Paramètres de Redis OSS 6
- Paramètres spécifiques au type de nœud MemoryDB

Modifications des paramètres de Redis OSS 7

> **Note**
>
> MemoryDB a introduit une version préliminaire de Vector Search qui inclut un nouveau groupe de paramètres immuables. `default.memorydb-redis7.search.preview` Ce groupe de paramètres est disponible dans la console MemoryDB et lors de la création d'un nouveau vector-search-enabled cluster à l'aide de la commande create-cluster CLI. La version préliminaire est disponible dans les AWS régions suivantes : USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Asie-Pacifique (Tokyo) et Europe (Irlande).

Famille de groupes de paramètres : memorydb_redis7

Les paramètres ajoutés dans Redis OSS 7 sont les suivants.

Nom	Détails	Description
`latency-tracking`	Valeurs autorisées : `yes`, `no` Par défaut : `no` Type : chaîne	Lorsque ce paramètre est défini sur yes, il suit les latences par commande et permet d'exporter la distribution percentile via la commande de statistiques de latence `INFO` et les distribut

Nom	Détails	Description
	Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	ions de latence cumulées (histogrammes) via la commande `LATENCY`.
`hash-max-listpack-entries`	Valeurs autorisées : `0+` Par défaut : `512` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Nombre maximum d'entrées de hachage pour que le jeu de données soit compressé.
`hash-max-listpack-value`	Valeurs autorisées : `0+` Par défaut : `64` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Le seuil des entrées de hachage les plus importantes pour que le jeu de données soit compressé.

Nom	Détails	Description
`zset-max-listpack-entries`	Valeurs autorisées : `0+` Par défaut : `128` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Nombre maximum d'entrées de jeu triées pour que le jeu de données soit compressé.
`zset-max-listpack-value`	Valeurs autorisées : `0+` Par défaut : `64` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Le seuil des entrées de jeu triées les plus importantes pour que le jeu de données soit compressé.

Nom	Détails	Description
`search-enabled`	Valeurs autorisées : `yes, no` Par défaut : no Type : chaîne Modifiable : oui Les modifications prennent effet : pour les nouveaux clusters uniquement. Version minimale du moteur : 7.1	Lorsqu'il est défini sur Oui, il active les fonctionnalités de recherche.
`search-query-timeout-ms`	Valeurs autorisées : `1 - 60,000` Par défaut : `10,000` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster. Version minimale du moteur : 7.1	Durée maximale en millisecondes pendant laquelle une requête de recherche est autorisée à s'exécuter.

Les paramètres modifiés dans Redis OSS 7 sont les suivants.

Nom	Détails	Description
`activerehashing`	Modifiable : no. Dans Redis OSS 7, ce paramètre est masqué et activé par défaut. Pour le désactiver, vous devez créer un cas de support.	Modifiable était sur oui.

Les paramètres supprimés dans Redis OSS 7 sont les suivants.

Nom	Détails	Description
`hash-max-ziplist-entries`	Valeurs autorisées : `0+` Par défaut : `512` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Utilisez `listpack` plutôt que `ziplist` pour représenter un petit encodage à hachage
`hash-max-ziplist-value`	Valeurs autorisées : `0+` Par défaut : `64` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Utilisez `listpack` plutôt que `ziplist` pour représenter un petit encodage à hachage

Nom	Détails	Description
`zset-max-ziplist-entries`	Valeurs autorisées : `0+` Par défaut : `128` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Utilisez `listpack` plutôt que `ziplist` pour représenter un petit encodage à hachage.
`zset-max-ziplist-value`	Valeurs autorisées : `0+` Par défaut : `64` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Utilisez `listpack` plutôt que `ziplist` pour représenter un petit encodage à hachage.

Paramètres de Redis OSS 6

> **Note**
>
> Dans la version 6.2 du moteur Redis OSS, lorsque la famille de nœuds r6gd a été introduite pour une utilisation avecMise à niveau des données, uniquementnoeviction, `volatile-lru` et les politiques de `allkeys-lru` mémoire maximale sont prises en charge avec les types de nœuds r6gd.

Famille de groupes de paramètres : memorydb_redis6

Les paramètres ajoutés dans Redis OSS 6 sont les suivants.

Nom	Détails	Description
`maxmemory-policy`	Type : CORDE Valeurs autorisées : volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu, volatile-random, allkeys-random, volatile-ttl, noeviction Par défaut : noeviction	La politique d'expulsion des clés lors de l'utilisation de la mémoire maximale est atteinte. Pour plus d'informations, voir Utilisation de Redis OSS comme cache LRU Utilisation de Redis OSS comme cache LRU.
`list-compress-depth`	Type : ENTIER Valeurs autorisées : 0- Par défaut : 0	La profondeur de compression correspond au nombre de nœuds des listes compressées et rapides de chaque côté de la liste à exclure de la compression. La tête et la queue de liste ne sont jamais compressées pour les opérations push et pop. Les paramètres sont : • 0 : Désactiver toute compression. • 1 : Commencer à compresser à partir du 1er nœud de la tête et de la queue. [tête]->nœud->nœud->...->nœud->[queue] Tous les nœuds sauf [tête] et [queue] sont compressés. • 2 : Commencer à compresser à partir du 2e nœud de la tête et de la queue. [tête]->[suivant]->nœud->nœud->...->nœud->[préc.]->[queue] [tête], [suivant], [préc.], [queue] ne pas compresser. Tous les autres nœuds sont compressés.

Nom	Détails	Description
		• Etc.
`hll-sparse-max-bytes`	Type : ENTIER Valeurs autorisées : 1 à 16 000 Par défaut: 3000	HyperLogLog limite d'octets de représentation clairsemée. La limite inclut l'en-tête de 16 octets. Lorsqu'une HyperLogLog représentation clairsemée dépasse cette limite, elle est convertie en représentation dense. Une valeur supérieure à 16 000 n'est pas recommandée car à ce stade, la représentation dense est plus efficace en termes de mémoire. Nous recommandons une valeur d'environ 3000 pour bénéficier des avantages d'un codage peu encombrant sans `PFADD` trop ralentir, ce qui correspond à O (N) avec un codage clairsemé. La valeur peut être portée à environ 10000 lorsque le processeur n'est pas un problème, mais que l'espace l'est, et que l'ensemble de données est composé de nombreuses données HyperLogLogs dont la cardinalité se situe entre 0 et 15 000.
`lfu-log-factor`	Type : ENTIER Valeurs autorisées : 1- Par défaut: 10	Facteur logarithmique utilisé pour incrémenter le compteur de clés dans le cadre de la politique d'expulsion de la LFU.
`lfu-decay-time`	Type : ENTIER Valeurs autorisées : 0- Valeur par défaut : 1	Le temps, en minutes, nécessaire pour décrémenter le compteur clé de la politique d'expulsion de la LFU.

Nom	Détails	Description
`active-defrag-max-scan-fields`	Type : ENTIER Valeurs autorisées : 1-1000000 Par défaut: 1000	Nombre maximum de champs set/hash/zset/list qui seront traités à partir de l'analyse du dictionnaire principal pendant la défragmentation active.
`active-defrag-threshold-upper`	Type : ENTIER Valeurs autorisées : 1-100 Par défaut : 100	Pourcentage maximum de fragmentation à partir duquel nous utilisons l'effort maximum.
`client-output-buffer-limit-pubsub-hard-limit`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 33554432	Pour les clients de publication/abonnement Redis OSS : si la mémoire tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté.
`client-output-buffer-limit-pubsub-soft-limit`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 8388608	Pour les clients de publication/abonnement Redis OSS : si la mémoire tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté, mais uniquement si cette condition persiste pendant `client-output-buffer-limit-pubsub-soft-seconds.`
`client-output-buffer-limit-pubsub-soft-seconds`	Type : ENTIER Valeurs autorisées : 0- Par défaut : 60	Pour les clients de publication/abonnement Redis OSS : si la mémoire tampon de sortie d'un client reste en `client-output-buffer-limit-pubsub-soft-limit` octets pendant plus de secondes, le client sera déconnecté.

Nom	Détails	Description
`timeout`	Type : ENTIER Valeurs autorisées : 0,20- Par défaut : 0	Le nombre de secondes qu'un nœud doit attendre avant d'être mis hors service. Les valeurs sont les suivantes : • 0 : ne déconnectez jamais un client inactif. • 1-19 : valeurs non valides. • >=20 : le nombre de secondes pendant lesquelles un nœud attend avant de déconnecter un client inactif.
`notify-keyspace-events`	Type : CORDE Valeurs autorisées : NULL Par défaut : NULL	Les événements keyspace dont Redis OSS doit informer les clients Pub/Sub. Par défaut, toutes les notifications sont désactivées.
`maxmemory-samples`	Type : ENTIER Valeurs autorisées : 1- Valeur par défaut : 3	Pour les `time-to-live (TTL)` calculs least-recently-used `(LRU)` et les calculs, ce paramètre représente la taille de l'échantillon de clés à vérifier. Par défaut, Redis OSS choisit 3 clés et utilise celle qui a été utilisée le moins récemment.
`slowlog-max-len`	Type : ENTIER Valeurs autorisées : 0- Valeur par défaut : 128	La longueur maximale du journal lent de Redis OSS. Il n'y a pas de limite à cette longueur. Sachez simplement que cela consommera de la mémoire. Vous pouvez récupérer la mémoire utilisée par le slow log avec `SLOWLOG RESET.`

Nom	Détails	Description
activerehashing	Type : CORDE Valeurs autorisées : oui, non Par défaut : oui	La table de hachage principal est répétée dix fois par seconde ; chaque nouvelle opération de hachage utilise 1 milliseconde de la durée d'utilisation de l'UC. Cette valeur est définie lorsque vous créez le groupe de paramètres. Lorsque vous assignez un nouveau groupe de paramètres à un cluster, cette valeur doit être le même dans l'ancien et dans le nouveau groupe de paramètres.
client-output-buffer-limit-normal-hard-limit	Type : ENTIER Valeurs autorisées : 0- Par défaut : 0	Si la mémoire tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté. La valeur par défaut est zéro (aucune limite stricte).
client-output-buffer-limit-normal-soft-limit	Type : ENTIER Valeurs autorisées : 0- Par défaut : 0	Si la mémoire tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté, mais uniquement si cette condition persiste pour client-output-buffer-limit-normal-soft-seconds . La valeur par défaut est zéro (aucune limite flexible).
client-output-buffer-limit-normal-soft-seconds	Type : ENTIER Valeurs autorisées : 0- Par défaut : 0	Si la mémoire tampon de sortie d'un client reste à client-output-buffer-limit-normal-soft-limit octets plus longtemps que ce nombre de secondes, le client sera déconnecté. La valeur par défaut est zéro (aucune limite de temps).

Nom	Détails	Description
`tcp-keepalive`	Type : ENTIER Valeurs autorisées : 0- Valeur par défaut : 300	Si la valeur est une valeur différente de zéro (N), les clients de nœud sont interrogés toutes les N secondes pour s'assurer qu'ils sont toujours connectés. Avec le paramètre par défaut de 0, aucune interrogation de ce type ne se produit.
`active-defrag-cycle-min`	Type : ENTIER Valeurs autorisées : 1-75 Par défaut: 5	Effort minimum pour défragmenter en pourcentage d'UC.
`stream-node-max-bytes`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 4096	La structure des données du flux est une arborescence de nœuds radix qui encodent plusieurs éléments à l'intérieur. Utilisez cette configuration pour spécifier la taille maximale d'un nœud unique dans une arborescence radix, exprimée en octets. Si la taille du nœud de l'arborescence est définie sur 0, elle n'est pas limitée.
`stream-node-max-entries`	Type : ENTIER Valeurs autorisées : 0- Par défaut : 100	La structure des données du flux est une arborescence de nœuds radix qui encodent plusieurs éléments à l'intérieur. Utilisez cette configuration pour spécifier le nombre maximal d'éléments que peut contenir un même nœud avant le basculement sur un nouveau nœud lors de l'ajout de nouvelles entrées de flux. S'il est défini sur 0, le nombre d'éléments dans le nœud de l'arborescence est illimité.

Nom	Détails	Description
`lazyfree-lazy-eviction`	Type : CORDE Valeurs autorisées : oui, non Par défaut : non	Effectuez une suppression asynchrone lors des expulsions.
`active-defrag-ignore-bytes`	Type : ENTIER Valeurs autorisées : 1048576- Par défaut: 104857600	Quantité minimum de fragmentation perdue pour lancer une défragmentation active.
`lazyfree-lazy-expire`	Type : CORDE Valeurs autorisées : oui, non Par défaut : non	Effectuez une suppression asynchrone sur les clés expirées.
`active-defrag-threshold-lower`	Type : ENTIER Valeurs autorisées : 1-100 Par défaut: 10	Pourcentage minimum de fragmentation pour lancer une défragmentation active.
`active-defrag-cycle-max`	Type : ENTIER Valeurs autorisées : 1-75 Par défaut: 75	Effort maximum pour défragmenter en pourcentage d'UC.
`lazyfree-lazy-server-del`	Type : CORDE Valeurs autorisées : oui, non Par défaut : non	Effectue une suppression asynchrone des commandes qui mettent à jour les valeurs.

Nom	Détails	Description
`slowlog-log-slower-than`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 10000	Le temps d'exécution maximal, en microsecondes, à dépasser pour que la commande soit enregistrée par la fonctionnalité Redis OSS`Slow Log`. Notez qu'un nombre négatif désactive le journal lent, tandis qu'une valeur de zéro force l'enregistrement de chaque commande.
`hash-max-ziplist-entries`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 512	Détermine la quantité de mémoire utilisée pour les fonctions de hachage. Les fonctions de hachage avec un nombre d'entrées inférieur à celui spécifié sont stockées à l'aide d'un encodage spécial qui économise de l'espace.
`hash-max-ziplist-value`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 64	Détermine la quantité de mémoire utilisée pour les fonctions de hachage. Les fonctions de hachage avec un nombre d'octets plus petit que le nombre spécifié sont stockées à l'aide d'un encodage spécial qui économise de l'espace.
`set-max-intset-entries`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 512	Détermine que la quantité de mémoire utilisée pour certains types de jeux (chaînes qui sont des entiers de base 10 dans la plage d'entiers signés de 64 bits). De tels jeux avec un nombre d'entrées inférieur à celui spécifié sont stockées à l'aide d'un encodage spécial qui économise de l'espace.

Nom	Détails	Description
`zset-max-ziplist-entries`	Type : ENTIER Valeurs autorisées : 0- Valeur par défaut : 128	Détermine la quantité de mémoire utilisée pour les jeux triés. Les jeux triés avec un nombre d'éléments inférieur à celui spécifié sont stockés à l'aide d'un encodage spécial qui économise de l'espace.
`zset-max-ziplist-value`	Type : ENTIER Valeurs autorisées : 0- Par défaut: 64	Détermine la quantité de mémoire utilisée pour les jeux triés. Les jeux triés avec des entrées qui ont un nombre d'octets plus petit que le nombre spécifié sont stockés à l'aide d'un encodage spécial qui économise de l'espace.
`tracking-table-max-keys`	Type : ENTIER Valeurs autorisées : 1-100000000 Par défaut : 1000000	Pour faciliter la mise en cache côté client, Redis OSS permet de suivre quels clients ont accédé à quelles clés. Lorsque la clé suivie est modifiée, des messages d'invalidation sont envoyés à tous les clients pour les avertir que leurs valeurs mises en cache ne sont plus valides. Cette valeur vous permet de spécifier la limite supérieure de cette table.
`acllog-max-len`	Type : ENTIER Valeurs autorisées : 1 à 10000 Valeur par défaut : 128	Le nombre maximum d'entrées dans le journal ACL.

Nom	Détails	Description
active-expire-effort	Type : ENTIER Valeurs autorisées : 1 à 10 Valeur par défaut : 1	Redis OSS supprime les clés qui ont dépassé leur durée de vie par deux mécanismes. Dans l'un, une clé est accessible et a expiré. Dans l'autre, un travail périodique échantillonne les clés et provoque l'expiration de celles qui ont dépassé leur time-to-live. Ce paramètre définit l'effort déployé par Redis OSS pour faire expirer les éléments du travail périodique. La valeur par défaut de 1 tente d'éviter que plus de 10 % des clés expirées restent en mémoire. Il essaie également d'éviter de consommer plus de 25 % de la mémoire totale et d'ajouter une latence au système. Vous pouvez augmenter cette valeur jusqu'à 10 pour augmenter l'effort consacré aux clés d'expiration. Le compromis est une utilisation CPU plus élevée et une latence potentiellement plus élevée. Nous recommandons une valeur de 1, sauf si vous constatez une utilisation élevée de la mémoire et que vous pouvez tolérer une augmentation de l'utilisation du processeur.
lazyfree-lazy-user-del	Type : CORDE Valeurs autorisées : oui, non Par défaut : non	Spécifie si le comportement par défaut de la `DEL` commande agit de la même manière que`UNLINK`.
activedefrag	Type : CORDE Valeurs autorisées : oui, non Par défaut : non	Défragmentation active de la mémoire activée.

Nom	Détails	Description
`maxclients`	Type : ENTIER Valeurs autorisées : 65000 Par défaut: 65000	Le nombre maximum de clients qui peut être connecté à un moment donné. Non modifiable.
`client-query-buffer-limit`	Type : ENTIER Valeurs autorisées : 1048576-1073741824 Par défaut: 1073741824	Taille maximum d'un seul tampon de requête client. Le changement a lieu immédiatement.
`proto-max-bulk-len`	Type : ENTIER Valeurs autorisées : 1048576-536870912 Par défaut: 536870912	Taille maximum d'une seule demande d'élément. Le changement a lieu immédiatement.

Paramètres spécifiques au type de nœud MemoryDB

Bien que la plupart des paramètres ont une valeur unique, certains paramètres ont des valeurs différentes en fonction du type de nœud utilisé. Le tableau suivant indique la valeur par défaut `maxmemory` pour chaque type de nœud. La valeur de `maxmemory` est le nombre maximal d'octets disponibles que vous pouvez utiliser pour les données et d'autres utilisations, sur le nœud.

Type de nœud	Maxmemory
db.r7g.large	14037181030
db.r7g.xlarge	28261849702
db.r7g.2xlarge	56711183565

Type de nœud	Maxmemory
db.r7g.4xlarge	113609865216
db.r7g.8xlarge	225000375228
db.r7g.12xlarge	341206346547
db.r7g.16xlarge	450000750456
db.r6gd.xlarge	28261849702
db.r6g.2xlarge	56711183565
db.r6g.4xlarge	113609865216
db.r6g.8xlarge	225000375228
db.r6g.large	14037181030
db.r6g.xlarge	28261849702
db.r6g.2xlarge	56711183565
db.r6g.4xlarge	113609865216
db.r6g.8xlarge	225000375228
db.r6g.12xlarge	341206346547
db.r6g.16xlarge	450000750456
db.t4g.small	1471026299
db.t4g.medium	3317862236

> **Note**
>
> Tous les types d'instances MemoryDB doivent être créés dans un VPC Amazon Virtual Private Cloud.

Tutoriel : Configuration d'une fonction Lambda pour accéder à MemoryDB dans un Amazon VPC

Dans ce didacticiel, vous apprendrez à :

- Créez un cluster MemoryDB dans votre Amazon Virtual Private Cloud (Amazon VPC) par défaut dans la région us-east-1.
- Créez une fonction Lambda pour accéder au cluster. Lorsque vous créez la fonction Lambda, vous fournissez des ID de sous-réseau dans votre VPV Amazon, ainsi qu'un groupe de sécurité de VPV pour permettre à la fonction Lambda d'accéder aux ressources dans votre VPC. À titre d'illustration dans ce didacticiel, la fonction Lambda génère un UUID, l'écrit dans le cluster et le récupère du cluster.
- Appelez la fonction Lambda manuellement et vérifiez qu'elle a accédé au cluster dans votre VPC.
- Nettoyez la fonction Lambda, le cluster et le rôle IAM configurés pour ce didacticiel.

Rubriques

- Étape 1 : créer un cluster
- Étape 2 : créer une fonction Lambda
- Étape 3 : Tester la fonction Lambda
- Étape 4 : Nettoyage (facultatif)

Étape 1 : créer un cluster

Pour créer un cluster, procédez comme suit.

Rubriques

- Étape 1.1 : Création d'un cluster
- Étape 1.2 : Copier le point de terminaison du cluster
- Étape 1.3 : Création d'un rôle IAM
- Étape 1.4 : Création d'une liste de contrôle d'accès (ACL)

Étape 1.1 : Création d'un cluster

Au cours de cette étape, vous créez un cluster dans le VPC Amazon par défaut de la région us-east-1 de votre compte à l'aide de la (CLI). AWS Command Line Interface Pour plus d'informations sur la création d'un cluster à l'aide de la console ou de l'API MemoryDB, voir. Étape 1 : créer un cluster

```
aws memorydb create-cluster --cluster-name cluster-01 --engine-version 7.0 --acl-name
 open-access \
--description "MemoryDB IAM auth application" \
--node-type db.r6g.large
```

Notez que la valeur du champ Statut est définie sur `CREATING`. MemoryDB peut prendre quelques minutes pour terminer la création de votre cluster.

Étape 1.2 : Copier le point de terminaison du cluster

Vérifiez que MemoryDB a terminé de créer le cluster à l'aide de la `describe-clusters` commande.

```
aws memorydb describe-clusters \
--cluster-name cluster-01
```

Copiez l'adresse du point de terminaison du cluster indiquée dans la sortie. Vous aurez besoin de cette adresse lorsque vous allez créer le package de déploiement de votre fonction Lambda.

Étape 1.3 : Création d'un rôle IAM

1. Créez un document de stratégie d'approbation IAM, comme indiqué ci-dessous, pour votre rôle afin d'autoriser votre compte à assumer le nouveau rôle. Enregistrez la politique dans un fichier nommé trust-policy.json. Assurez-vous de remplacer account_id 123456789012 dans cette politique par votre account_id.

   ```
   {
   "Version": "2012-10-17",
       "Statement": [{
           "Effect": "Allow",
           "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
           "Action": "sts:AssumeRole"
       },
   ```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }]
}
```

2. Créez un document de politique IAM, comme indiqué ci-dessous. Enregistrez la politique dans un fichier nommé policy.json. Assurez-vous de remplacer account_id 123456789012 dans cette politique par votre account_id.

```
{
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "memorydb:Connect"
      ],
      "Resource" : [
        "arn:aws:memorydb:us-east-1:123456789012:cluster/cluster-01",
        "arn:aws:memorydb:us-east-1:123456789012:user/iam-user-01"
      ]
    }
  ]
}
```

3. Créez un rôle IAM.

```
aws iam create-role \
--role-name "memorydb-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. Créez la politique IAM.

```
aws iam create-policy \
  --policy-name "memorydb-allow-all" \
  --policy-document file://policy.json
```

5. Attachez la politique gérée IAM au rôle. Assurez-vous de remplacer account_id 123456789012 dans ce policy-arn par votre account_id.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Étape 1.4 : Création d'une liste de contrôle d'accès (ACL)

1. Créez un compte utilisateur prenant en charge IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
--authentication-mode Type=iam \
--access-string "on ~* +@all"
```

2. Créez une ACL et attachez-la au cluster.

```
aws memorydb create-acl \
  --acl-name iam-acl-01 \
  --user-names iam-user-01

aws memorydb update-cluster \
  --cluster-name cluster-01 \
  --acl-name iam-acl-01
```

Étape 2 : créer une fonction Lambda

Pour créer une fonction Lambda, procédez comme suit.

Rubriques

- Étape 2.1 : Créer le package de déploiement
- Étape 2.2 : Créer un rôle IAM (rôle d'exécution)
- Étape 2.3 : Charger le package de déploiement (créer la fonction Lambda)

Étape 2.1 : Créer le package de déploiement

Dans ce didacticiel, nous fournissons un exemple de code en Python pour votre fonction Lambda.

Python

L'exemple de code Python suivant lit et écrit un élément dans votre cluster MemoryDB. Copiez le code et enregistrez-le dans un fichier nommé `app.py`. Assurez-vous de remplacer la `cluster_endpoint` valeur du code par l'adresse du point de terminaison que vous avez copiée à l'étape 1.2.

```
from typing import Tuple, Union
from urllib.parse import ParseResult, urlencode, urlunparse

import botocore.session
import redis
from botocore.model import ServiceId
from botocore.signers import RequestSigner
from cachetools import TTLCache, cached
import uuid

class MemoryDBIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cluster_name, region="us-east-1"):
        self.user = user
        self.cluster_name = cluster_name
        self.region = region

        session = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("memorydb"),
            self.region,
            "memorydb",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )

    # Generated IAM tokens are valid for 15 minutes
    @cached(cache=TTLCache(maxsize=128, ttl=900))
    def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
        query_params = {"Action": "connect", "User": self.user}

        url = urlunparse(
            ParseResult(
                scheme="https",
                netloc=self.cluster_name,
                path="/",
                query=urlencode(query_params),
                params="",
```

```
                fragment="",
            )
        )
        signed_url = self.request_signer.generate_presigned_url(
            {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
            operation_name="connect",
            expires_in=900,
            region_name=self.region,
        )
        # RequestSigner only seems to work if the URL has a protocol, but
        # MemoryDB only accepts the URL without a protocol
        # So strip it off the signed URL before returning
        return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cluster_name = "cluster-01" # replace with your cache name
    cluster_endpoint = "clustercfg.cluster-01.xxxxxx.memorydb.us-east-1.amazonaws.com"
 # replace with your cluster endpoint
    creds_provider = MemoryDBIAMProvider(user=username, cluster_name=cluster_name)
    redis_client = redis.Redis(host=cluster_endpoint, port=6379,
 credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cluster
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cluster and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from MemoryDB.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
 {decoded_result}")

    return "Fetched value from MemoryDB"
```

Ce code utilise la `redis-py` bibliothèque Python pour placer des éléments dans votre cluster et les récupérer. Ce code est utilisé `cachetools` pour mettre en cache les jetons d'authentification IAM générés pendant 15 minutes. Pour créer un package de déploiement contenant `redis-py` et`cachetools`, effectuez les étapes suivantes.

Dans le répertoire de votre projet contenant le fichier de code `app.py` source, créez un package de dossiers dans lequel installer les `cachetools` bibliothèques `redis-py` et.

```
mkdir package
```

Installez `redis-py` et `cachetools` utilisez pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Créez un fichier .zip contenant les `cachetools` bibliothèques `redis-py` et. Sous Linux et macOS, exécutez la commande suivante. Sous Windows, utilisez l'utilitaire zip de votre choix pour créer un fichier .zip avec les `cachetools` bibliothèques `redis-py` et à la racine.

```
cd package
zip -r ../my_deployment_package.zip
```

Ajoutez votre code de fonction dans le fichier .zip. Sous Linux et macOS, exécutez la commande suivante. Sous Windows, utilisez l'utilitaire zip de votre choix pour ajouter le fichier app.py à la racine de votre fichier .zip.

```
cd ..
zip my_deployment_package.zip app.py
```

Étape 2.2 : Créer un rôle IAM (rôle d'exécution)

Attachez la politique AWS gérée nommée `AWSLambdaVPCAccessExecutionRole` au rôle.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"
```

Étape 2.3 : Charger le package de déploiement (créer la fonction Lambda)

Dans cette étape, vous créez la fonction Lambda (AccessMemoryDB) à l'aide de la commande AWS CLI create-function.

Dans le répertoire du projet qui contient le fichier .zip de votre package de déploiement, exécutez la commande Lambda `create-function` CLI suivante.

Pour l'option de rôle, utilisez l'ARN du rôle d'exécution que vous avez créé à l'étape 2.2. Pour le vpc-config, entrez des listes séparées par des virgules des sous-réseaux de votre VPC par défaut et de l'ID du groupe de sécurité de votre VPC par défaut. Vous trouverez ces valeurs dans la console Amazon VPC. Pour trouver les sous-réseaux de votre VPC par défaut, sélectionnez Vos VPC, puis choisissez le VPC par défaut AWS de votre compte. Pour trouver le groupe de sécurité pour ce VPC, accédez à Sécurité et choisissez Groupes de sécurité. Assurez-vous que la région us-east-1 est sélectionnée.

```
aws lambda create-function \
--function-name AccessMemoryDB  \
--region us-east-1 \
--zip-file fileb://my_deployment_package.zip \
--role arn:aws:iam::123456789012:role/memorydb-iam-auth-app \
--handler app.lambda_handler \
--runtime python3.12  \
--timeout 30 \
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-
security-group-id
```

Étape 3 : Tester la fonction Lambda

Au cours de cette étape, vous invoquez la fonction Lambda manuellement à l'aide de la commande invoke. Lorsque la fonction Lambda s'exécute, elle génère un UUID et l'écrit dans le ElastiCache cache que vous avez spécifié dans votre code Lambda. La fonction Lambda récupère ensuite l'élément à partir du cache.

1. Appelez la fonction Lambda (AccessMemoryDB) à l'aide de la commande AWS Lambda invoke.

   ```
   aws lambda invoke \
   --function-name AccessMemoryDB  \
   --region us-east-1 \
   output.txt
   ```

2. Vérifiez que l'exécution de la fonction Lambda a réussi comme suit :

   - Passez en revue le fichier output.txt.
   - Vérifiez les résultats dans CloudWatch Logs en ouvrant la CloudWatch console et en choisissant le groupe de journaux pour votre fonction (AccessRedis/aws/lambda/). Le flux de journaux doit contenir un résultat similaire à ce qui suit :

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched
 826e70c5f4d2478c8c18027125a3e01e from MemoryDB.
```

- Passez en revue les résultats dans la AWS Lambda console.

Étape 4 : Nettoyage (facultatif)

Pour nettoyer, procédez comme suit.

Rubriques

- Étape 4.1 : Supprimer la fonction Lambda
- Étape 4.2 : Supprimer le cluster MemoryDB
- Étape 4.3 : Supprimer le rôle et les politiques IAM

Étape 4.1 : Supprimer la fonction Lambda

```
aws lambda delete-function \
 --function-name AccessMemoryDB
```

Étape 4.2 : Supprimer le cluster MemoryDB

Supprimez le cluster.

```
aws memorydb delete-cluster \
 --cluster-name cluster-01
```

Supprimez l'utilisateur et l'ACL.

```
aws memorydb delete-user \
 --user-id iam-user-01

aws memorydb delete-acl \
 --acl-name iam-acl-01
```

Étape 4.3 : Supprimer le rôle et les politiques IAM

```
aws iam detach-role-policy \
```

```
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"

aws iam detach-role-policy \
--role-name "memorydb-iam-auth-app" \
--policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"

aws iam delete-role \
 --role-name "memorydb-iam-auth-app"

 aws iam delete-policy \
  --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Recherche vectorielle

La recherche vectorielle pour MemoryDB étend les fonctionnalités de MemoryDB. La recherche vectorielle peut être utilisée conjointement avec les fonctionnalités existantes de MemoryDB. Les applications qui n'utilisent pas la recherche vectorielle ne sont pas affectées par sa présence. La recherche vectorielle est disponible dans toutes les régions où MemoryDB est disponible.

La recherche vectorielle simplifie l'architecture de votre application tout en fournissant une recherche vectorielle à haut débit. La recherche vectorielle pour MemoryDB est idéale pour les cas d'utilisation où les performances optimales et l'évolutivité sont les critères de sélection les plus importants. Vous pouvez utiliser vos données MemoryDB ou Redis existantes OSS API pour créer des cas d'utilisation de l'apprentissage automatique et de l'IA générative, tels que la génération augmentée par extraction, la détection d'anomalies, la récupération de documents et les recommandations en temps réel.

Depuis le 26/06/2024, AWS MemoryDB offre les performances de recherche vectorielle les plus rapides avec les taux de rappel les plus élevés parmi les bases de données vectorielles les plus populaires sur. AWS

Rubriques

- Aperçu de la recherche vectorielle
- Cas d’utilisation
- Caractéristiques et limites de la recherche vectorielle
- En utilisant le AWS Management Console
- En utilisant le AWS Command Line Interface
- Commandes de recherche vectorielle

Aperçu de la recherche vectorielle

La recherche vectorielle repose sur la création, la maintenance et l'utilisation d'index. Chaque opération de recherche vectorielle spécifie un seul index et son opération est limitée à cet index, c'est-à-dire que les opérations sur un index ne sont pas affectées par les opérations sur un autre index. À l'exception des opérations de création et de destruction d'index, un certain nombre d'opérations peuvent être effectuées sur n'importe quel index à tout moment, ce qui signifie qu'au niveau du cluster, plusieurs opérations sur plusieurs index peuvent être en cours simultanément.

Les index individuels sont des objets nommés qui existent dans un espace de noms unique, distinct des autres OSS espaces de noms Redis : clés, fonctions, etc. Chaque index est conceptuellement similaire à une table de base de données classique dans la mesure où il est structuré en deux dimensions : colonne et lignes. Chaque ligne du tableau correspond à une OSS clé Redis. Chaque colonne de l'index correspond à un membre ou à une partie de cette clé. Dans ce document, les termes clé, ligne et enregistrement sont identiques et utilisés de manière interchangeable. De même, les termes colonne, champ, chemin et membre sont essentiellement identiques et sont également utilisés de manière interchangeable.

Il n'existe aucune commande spéciale pour ajouter, supprimer ou modifier des données indexées. Au contraire, JSON les commandes existantes HASH ou qui modifient une clé figurant dans un index mettent également automatiquement à jour l'index.

Rubriques

- Les index et le keyspace Redis OSS
- Types de champs d'index
- Algorithmes d'index vectoriel
- Expression de requête de recherche vectorielle
- INFO commande
- Sécurité de la recherche vectorielle

Les index et le keyspace Redis OSS

Les index sont construits et gérés sur un sous-ensemble de l'espace de touches RedisOSS. Plusieurs index peuvent choisir des sous-ensembles disjoints ou superposés de l'espace de touches Redis OSS sans limitation. L'espace-clé de chaque index est défini par une liste de préfixes clés fournis lors de la création de l'index. La liste des préfixes est facultative et si elle est omise, l'OSSespace clé Redis entier fera partie de cet index. Les index sont également saisis dans la mesure où ils ne couvrent que les clés dont le type correspond. Actuellement, seuls JSON les HASH index sont pris en charge. Un HASH index indexe uniquement les HASH clés couvertes par sa liste de préfixes. De même, un JSON index indexe uniquement les JSON clés couvertes par sa liste de préfixes. Les clés de la liste de préfixes d'espaces de touches d'un index qui n'ont pas le type désigné sont ignorées et n'affectent pas les opérations de recherche.

Lorsqu'une JSON commande HASH ou modifie une touche qui se trouve dans un espace clé d'un index, cet index est mis à jour. Ce processus consiste à extraire les champs déclarés pour chaque

index et à mettre à jour l'index avec la nouvelle valeur. Le processus de mise à jour est effectué dans un fil de discussion en arrière-plan, ce qui signifie que les index ne sont cohérents qu'en fin de compte avec le contenu de leur espace clavier. Ainsi, l'insertion ou la mise à jour d'une clé ne sera pas visible dans les résultats de recherche pendant une courte période. Pendant les périodes de forte charge du système et/ou de forte mutation des données, le délai de visibilité peut s'allonger.

La création d'un index est un processus en plusieurs étapes. La première étape consiste à exécuter le FT. CREATEcommande qui définit l'index. L'exécution réussie d'une création initie automatiquement la deuxième étape : le remblayage. Le processus de remplissage s'exécute dans un fil d'arrière-plan et scanne l'espace OSS clé Redis à la recherche de clés figurant dans la liste de préfixes du nouvel index. Chaque clé trouvée est ajoutée à l'index. Finalement, l'espace clé entier est scanné, complétant ainsi le processus de création de l'index. Notez que lorsque le processus de remplissage d'index est en cours d'exécution, les mutations de clés indexées sont autorisées, il n'y a aucune restriction et le processus de remplissage d'index ne sera pas terminé tant que toutes les clés ne seront pas correctement indexées. Les opérations de requête tentées alors qu'un index est en cours de remblayage ne sont pas autorisées et se terminent par une erreur. L'achèvement du processus de remblayage peut être déterminé à partir de la sortie de la `FT.INFO` commande pour cet index ('backfill_status').

Types de champs d'index

Chaque champ (colonne) d'un index possède un type spécifique déclaré lors de la création de l'index et un emplacement dans une clé. Pour HASH les clés, l'emplacement est le nom du champ dans leHASH. Pour JSON les clés, l'emplacement est une description du JSON chemin. Lorsqu'une clé est modifiée, les données associées aux champs déclarés sont extraites, converties dans le type déclaré et stockées dans l'index. Si les données sont manquantes ou ne peuvent pas être converties correctement dans le type déclaré, ce champ est omis de l'index. Il existe quatre types de champs, comme expliqué ci-dessous :

- Les champs numériques contiennent un seul chiffre. Pour JSON les champs, les règles numériques des JSON nombres doivent être respectées. En HASH effet, le champ est censé contenir le ASCII texte d'un nombre écrit dans le format standard pour les nombres fixes ou à virgule flottante. Quelle que soit la représentation dans la clé, ce champ est converti en un nombre à virgule flottante de 64 bits pour être stocké dans l'index. Les champs numériques peuvent être utilisés avec l'opérateur de recherche par plage. Comme les nombres sous-jacents sont stockés en virgule flottante avec ses limites de précision, les règles habituelles relatives aux comparaisons numériques pour les nombres à virgule flottante s'appliquent.

- Les champs de balises contiennent zéro ou plusieurs valeurs de balise codées sous la forme d'une seule chaîne UTF -8. La chaîne est analysée en valeurs de balise à l'aide d'un caractère séparateur (la valeur par défaut est une virgule mais peut être remplacée), les espaces blancs de début et de fin étant supprimés. Un seul champ de balise peut contenir autant de valeurs de balise que vous le souhaitez. Les champs de balises peuvent être utilisés pour filtrer les requêtes relatives à l'équivalence des valeurs de balises par une comparaison entre majuscules et minuscules.
- Les champs de texte contiennent un blob d'octets qui n'ont pas besoin d'être conformes à la norme UTF -8. Les champs de texte peuvent être utilisés pour décorer les résultats des requêtes avec des valeurs pertinentes pour l'application. Par exemple, un URL ou le contenu d'un document, etc.
- Les champs vectoriels contiennent un vecteur de nombres, également appelé incorporation. Les champs vectoriels prennent en charge la recherche par K-plus proche voisin (KNN) de vecteurs de taille fixe à l'aide d'un algorithme et d'une métrique de distance spécifiés. Pour les HASH index, le champ doit contenir le vecteur entier codé au format binaire (IEEElittle-endian 754). Pour JSON les clés, le chemin doit faire référence à un tableau de la bonne taille rempli de chiffres. Notez que lorsqu'un JSON tableau est utilisé comme champ vectoriel, la représentation interne du tableau dans la JSON clé est convertie dans le format requis par l'algorithme sélectionné, ce qui réduit la consommation de mémoire et la précision. Les opérations de lecture ultérieures utilisant les JSON commandes produiront la valeur de précision réduite.

Algorithmes d'index vectoriel

Deux algorithmes d'index vectoriel sont fournis :

- Plat — L'algorithme Flat est un traitement linéaire par force brute de chaque vecteur de l'indice, fournissant des réponses exactes dans les limites de précision des calculs de distance. En raison du traitement linéaire de l'indice, les temps d'exécution de cet algorithme peuvent être très élevés pour les grands indices.
- HNSW(Petits mondes navigables hiérarchiques) — L'HNSWalgorithme est une alternative qui fournit une approximation de la bonne réponse en échange de temps d'exécution nettement plus courts. L'algorithme est contrôlé par trois paramètres`M`, `EF_CONSTRUCTION` et`EF_RUNTIME`. Les deux premiers paramètres sont spécifiés au moment de la création de l'index et ne peuvent pas être modifiés. Le `EF_RUNTIME` paramètre possède une valeur par défaut qui est spécifiée lors de la création de l'index, mais qui peut ensuite être remplacée lors de n'importe quelle opération de requête individuelle. Ces trois paramètres interagissent pour équilibrer la mémoire et la CPU

consommation lors des opérations d'ingestion et de requête, ainsi que pour contrôler la qualité de l'approximation d'une KNN recherche exacte (connue sous le nom de ratio de rappel).

Les deux algorithmes de recherche vectorielle (Flat etHNSW) prennent en charge un `INITIAL_CAP` paramètre facultatif. Lorsqu'il est spécifié, ce paramètre préalloue de la mémoire aux index, ce qui permet de réduire la charge de gestion de la mémoire et d'augmenter les taux d'ingestion de vecteurs.

Les algorithmes de recherche vectorielle tels que ceux-ci HNSW peuvent ne pas gérer efficacement la suppression ou le remplacement de vecteurs précédemment insérés. L'utilisation de ces opérations peut entraîner une consommation excessive de mémoire d'index et/ou une dégradation de la qualité du rappel. La réindexation est une méthode permettant de rétablir une utilisation et/ou un rappel optimaux de la mémoire.

Expression de requête de recherche vectorielle

Le FT. SEARCHet FT. AGGREGATEles commandes nécessitent une expression de requête. Cette expression est un paramètre de chaîne unique composé d'un ou de plusieurs opérateurs. Chaque opérateur utilise un champ de l'index pour identifier un sous-ensemble des clés de l'index. Plusieurs opérateurs peuvent être combinés à l'aide de combineurs booléens ainsi que de parenthèses pour améliorer ou restreindre davantage le jeu de clés collecté (ou le jeu de résultats).

Caractère générique

L'opérateur générique, l'astérisque (« * »), correspond à toutes les clés de l'index.

Plage numérique

La syntaxe de l'opérateur de plage numérique est la suivante :

```
<range-search> ::= '@' <numeric-field-name> ':' '[' <bound> <bound> ']'
<bound>  ::= <number> | '(' <number>
<number> ::= <integer> | <fixed-point> | <floating-point> | 'Inf' | '-Inf' | '+Inf'
```

Le < numeric-field-name > doit être un champ de type déclaré`NUMERIC`. Par défaut, la borne est inclusive, mais une parenthèse ouverte initiale ['('] peut être utilisée pour rendre une borne exclusive. La recherche par plage peut être convertie en une comparaison relationnelle unique (<, <=, >, >=) en utilisant `Inf` `+Inf` ou `-Inf` comme l'une des limites. Quel que soit le format numérique spécifié

(entier, virgule fixe, virgule flottante, infini), le nombre est converti en virgule flottante de 64 bits pour effectuer des comparaisons, réduisant ainsi la précision en conséquence.

Example Exemples

```
@numeric-field:[0 10]                    // 0   <= <value> <= 10
@numeric-field:[(0 10]                   // 0   <  <value> <= 10
@numeric-field:[0 (10]                   // 0   <= <value> <  10
@numeric-field:[(0 (10]                  // 0   <  <value> <  10
@numeric-field:[1.5 (Inf]                // 1.5 <= value
```

Tag : comparer

La syntaxe de l'opérateur de comparaison de balises est la suivante :

```
<tag-search> ::= '@' <tag-field-name> ':' '{' <tag> [ '|' <tag> ]* '}'
```

Si l'une des balises de l'opérateur correspond à l'une des balises du champ de balise de l'enregistrement, l'enregistrement est inclus dans le jeu de résultats. Le champ conçu par `<tag-field-name>` doit être un champ de l'index déclaré avec type`TAG`. Voici des exemples de comparaison de balises :

```
@tag-field:{ atag }
@tag-field: { tag1 | tag2 }
```

Combinaisons booléennes

Les ensembles de résultats d'un opérateur numérique ou d'un opérateur de balise peuvent être combinés à l'aide de la logique booléenne : et/ou. Les parenthèses peuvent être utilisées pour regrouper les opérateurs et/ou modifier l'ordre d'évaluation. La syntaxe des opérateurs de logique booléenne est la suivante :

```
<expression> ::= <phrase> | <phrase> '|' <expression> | '(' <expression> ')'
<phrase> ::= <term> | <term> <phrase>
<term> ::= <range-search> | <tag-search> | '*'
```

Plusieurs termes combinés dans une phrase sont « et ». Les phrases multiples combinées avec le tube (« | ») sont de type « ou ».

Recherche vectorielle

Les index vectoriels prennent en charge deux méthodes de recherche différentes : le voisin le plus proche et le range. Une recherche dans le plus proche voisin permet de localiser un certain nombre, K, des vecteurs de l'index les plus proches du vecteur (de référence) fourni. C'est ce que l'on appelle communément « K » voisins KNN les plus proches. La syntaxe d'une KNN recherche est la suivante :

```
<vector-knn-search> ::= <expression> '=>[KNN' <k> '@' <vector-field-name> '$'
 <parameter-name> <modifiers> ']'
<modifiers> ::= [ 'EF_RUNTIME' <integer> ] [ 'AS' <distance-field-name>]
```

Une KNN recherche vectorielle n'est appliquée qu'aux vecteurs qui répondent aux critères, `<expression>` qui peuvent être n'importe quelle combinaison des opérateurs définis ci-dessus : caractère générique, recherche par plage, recherche par étiquette et/ou combinaisons booléennes de ces derniers.

- <k>est un entier spécifiant le nombre de vecteurs les plus proches voisins à renvoyer.
- `<vector-field-name>`doit spécifier un champ de type déclaré`VECTOR`.
- `<parameter-name>`le champ indique l'une des entrées de la `PARAM` table de la `FT.AGGREGATE` commande `FT.SEARCH` ou. Ce paramètre est la valeur du vecteur de référence pour les calculs de distance. La valeur du vecteur est codée dans la `PARAM` valeur au format binaire little-endian IEEE 754 (même encodé que pour un champ vectoriel) HASH
- Pour les index vectoriels de typeHNSW, la `EF_RUNTIME` clause facultative peut être utilisée pour remplacer la valeur par défaut du `EF_RUNTIME` paramètre établi lors de la création de l'index.
- L'option `<distance-field-name>` fournit un nom de champ pour le jeu de résultats afin qu'il contienne la distance calculée entre le vecteur de référence et la clé localisée.

Une recherche par plage permet de localiser tous les vecteurs situés à une distance (rayon) spécifiée par rapport à un vecteur de référence. La syntaxe d'une recherche par plage est la suivante :

```
<vector-range-search> ::= ‘@’ <vector-field-name> ‘:’ ‘[’ ‘VECTOR_RANGE’ ( <radius> |
 ‘$’ <radius-parameter> )  $<reference-vector-parameter> ‘]’ [ ‘=’ ‘>’ ‘{’ <modifiers>
 ‘}’ ]
<modifiers> ::= <modifier> | <modifiers>, <modifier>
<modifer> ::= [ ‘$yield_distance_as’ ‘:’ <distance-field-name> ] [ ‘$epsilon’ ‘:’
 <epsilon-value> ]
```

Où :

- `<vector-field-name>`est le nom du champ vectoriel à rechercher.
- `<radius> or $<radius-parameter>`est la limite de distance numérique pour la recherche.
- `$<reference-vector-parameter>` est le nom du paramètre qui contient le vecteur de référence. La valeur du vecteur est codée dans la PARAM valeur au format binaire little-endian IEEE 754 (même encodage que pour un champ vectoriel) HASH
- L'option `<distance-field-name>` fournit un nom de champ pour le jeu de résultats afin qu'il contienne la distance calculée entre le vecteur de référence et chaque clé.
- L'option permet de `<epsilon-value>` contrôler les limites de l'opération de recherche, les vecteurs situés à distance `<radius> * (1.0 + <epsilon-value>)` sont parcourus à la recherche de résultats candidats. La valeur par défaut est 0,01.

INFO commande

La recherche vectorielle complète la OSS INFOcommande Redis avec plusieurs sections supplémentaires de statistiques et de compteurs. Une demande de récupération de la section `SEARCH` permet de récupérer toutes les sections suivantes :

`search_memory` Section

Name (Nom)	Description
search_used_memory_bytes	Nombre d'octets de mémoire consommés dans toutes les structures de données de recherche
search_used_memory_human	Version lisible par l'homme de ce qui précède

`search_index_stats` Section

Name (Nom)	Description
numéro_de_index_de_recherche	Nombre d'index créés
search_num_fulltext_indexes	Nombre de champs non vectoriels dans tous les index

Name (Nom)	Description
search_num_vector_indexes	Nombre de champs vectoriels dans tous les index
search_num_hash_indexes	Nombre d'index sur les touches HASH de saisie
search_num_json_indexes	Nombre d'index sur les touches JSON de saisie
search_total_indexed_keys	Nombre total de clés dans tous les index
search_total_indexed_vector	Nombre total de vecteurs dans tous les index
search_total_indexed_hash_keys	Nombre total de clés de type HASH dans tous les index
search_total_indexed_json_keys	Nombre total de clés de type JSON dans tous les index
search_total_index_size	Octets utilisés par tous les index
search_total_fulltext_index_size	Octets utilisés par les structures d'index non vectorielles
search_total_vector_index_size	Octets utilisés par les structures d'index vectoriels
search_max_index_lag_ms	Délai d'ingestion lors de la dernière mise à jour du lot d'ingestion

`search_ingestion` Section

Name (Nom)	Description
search_background_indexing_status	État de l'ingestion. `NO_ACTIVITY` signifie inactif. D'autres valeurs indiquent que des clés sont en cours d'ingestion.

Name (Nom)	Description
search_ingestion_paused	Sauf lors du redémarrage, cela doit toujours être « non ».

`search_backfill` Section

> **Note**
>
> Certains des champs documentés dans cette section ne sont visibles que lorsqu'un remblayage est en cours.

Name (Nom)	Description
search_num_active_backfills	Nombre d'activités de remblayage en cours
search_backfills_paused	Sauf en cas de manque de mémoire, cela doit toujours être « non ».
search_current_backfill_progress_percentage	% d'achèvement (0-100) du remblai actuel

`search_query` Section

Name (Nom)	Description
search_num_active_queries	Nombre de commandes en cours `FT.SEARCH` et `FT.AGGREGATE` commandes en cours

Sécurité de la recherche vectorielle

Les mécanismes de sécurité Redis OSS ACL (Access Control Lists) pour l'accès aux commandes et aux données sont étendus pour contrôler la fonction de recherche. ACLle contrôle des commandes de recherche individuelles est entièrement pris en charge. Une nouvelle ACL catégorie est fournie et de nombreuses catégories existantes (`@fast`,, `@read@write`, etc.) sont mises à jour pour inclure les

nouvelles commandes. `@search` Les commandes de recherche ne modifient pas les données clés, ce qui signifie que les ACL mécanismes existants pour l'accès en écriture sont préservés. Les règles d'accès HASH et les JSON opérations ne sont pas modifiées par la présence d'un index ; le contrôle d'accès normal au niveau des clés est toujours appliqué à ces commandes.

L'accès aux commandes de recherche avec index est également contrôlé via Redis OSSACL. Les contrôles d'accès sont effectués au niveau de l'index complet, et non au niveau de chaque clé. Cela signifie que l'accès à un index n'est accordé à un utilisateur que s'il est autorisé à accéder à toutes les clés possibles dans la liste des préfixes d'espace-touches de cet index. En d'autres termes, le contenu réel d'un index ne contrôle pas l'accès. C'est plutôt le contenu théorique d'un index tel que défini par la liste de préfixes qui est utilisé pour le contrôle de sécurité. Il peut être facile de créer une situation dans laquelle un utilisateur a accès en lecture et/ou en écriture à une clé mais ne peut pas accéder à un index contenant cette clé. Notez que seul l'accès en lecture au keyspace est requis pour créer ou utiliser un index ; la présence ou l'absence d'accès en écriture n'est pas prise en compte.

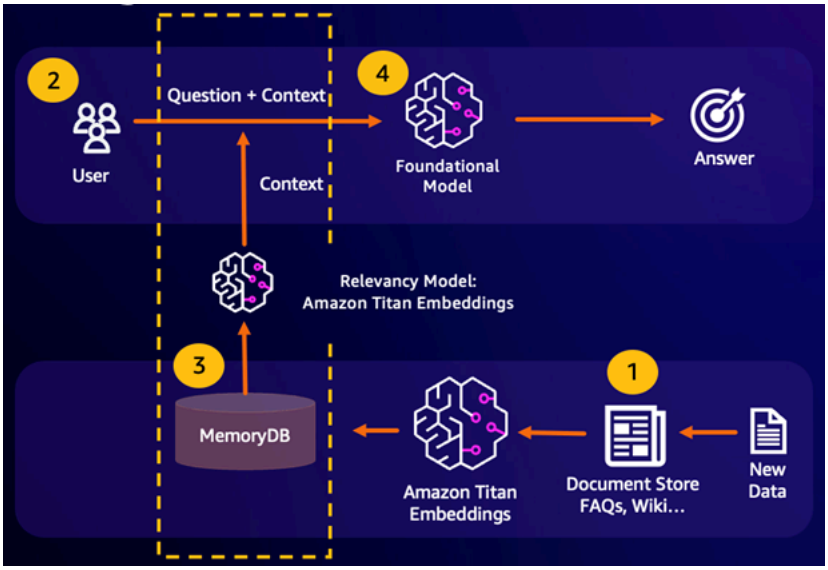
Pour plus d'informations sur l'utilisation ACLs de MemoryDB, voir Authentification des utilisateurs avec des listes de contrôle d'accès (). ACLs

Cas d’utilisation

Vous trouverez ci-dessous des cas d'utilisation de la recherche vectorielle.

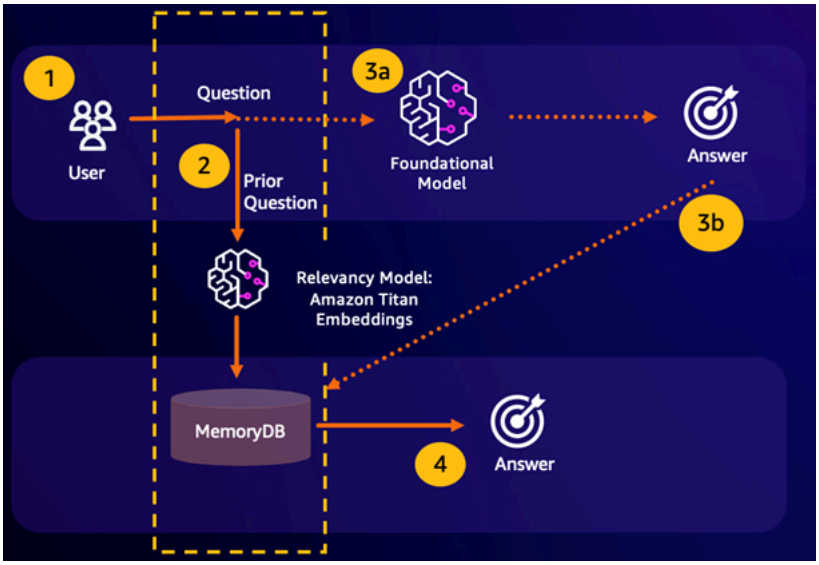
Génération augmentée de récupération () RAG

Retrieval Augmented Generation (RAG) utilise la recherche vectorielle pour récupérer des passages pertinents à partir d'un vaste corpus de données afin d'enrichir un grand modèle linguistique (). LLM Plus précisément, un encodeur intègre le contexte d'entrée et la requête de recherche dans des vecteurs, puis utilise une recherche approximative du plus proche voisin pour trouver des passages sémantiquement similaires. Ces passages récupérés sont concaténés avec le contexte d'origine pour fournir des informations pertinentes supplémentaires LLM afin de renvoyer une réponse plus précise à l'utilisateur.



Cache sémantique durable

La mise en cache sémantique est un processus visant à réduire les coûts de calcul en stockant les résultats précédents du FM. En réutilisant les résultats précédents issus d'inférences antérieures au lieu de les recalculer, la mise en cache sémantique réduit la quantité de calcul requise lors de l'inférence via le. FMs MemoryDB permet une mise en cache sémantique durable, ce qui évite la perte de données de vos inférences passées. Cela permet à vos applications d'IA générative de répondre en quelques millisecondes à un chiffre avec des réponses à des questions sémantiquement similaires antérieures, tout en réduisant les coûts en évitant les inférences inutiles. LLM



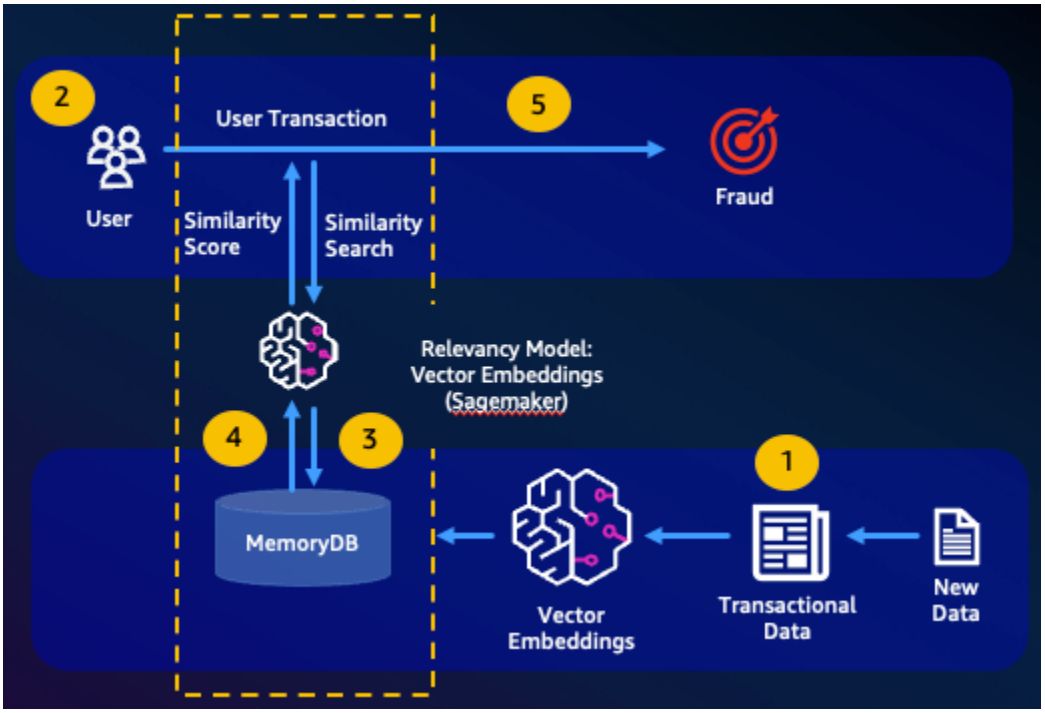
- Résultat de recherche sémantique : si la requête d'un client est sémantiquement similaire sur la base d'un score de similarité défini à une question précédente, la mémoire tampon FM

(MemoryDB) renverra la réponse à la question précédente à l'étape 4 et n'appellera pas le FM pendant les étapes 3. Cela permettra d'éviter la latence du modèle de base (FM) et les coûts encourus, offrant ainsi une expérience plus rapide au client.

- Erreur de recherche sémantique : si la requête d'un client n'est pas sémantiquement similaire, sur la base d'un score de similarité défini, à une requête précédente, le client appellera le FM pour lui fournir une réponse à l'étape 3a. La réponse générée par le FM sera ensuite stockée sous forme de vecteur dans MemoryDB pour les futures requêtes (étape 3b) afin de minimiser les coûts du FM sur des questions sémantiquement similaires. Dans ce flux, l'étape 4 ne serait pas invoquée car il n'y avait aucune question sémantiquement similaire pour la requête d'origine.

Détection des fraudes

La détection des fraudes, une forme de détection des anomalies, représente les transactions valides sous forme de vecteurs tout en comparant les représentations vectorielles des nouvelles transactions nettes. Une fraude est détectée lorsque ces nouvelles transactions nettes présentent une faible similitude avec les vecteurs représentant les données transactionnelles valides. Cela permet de détecter la fraude en modélisant un comportement normal, plutôt que d'essayer de prévoir tous les cas de fraude possibles. MemoryDB permet aux entreprises de le faire en période de débit élevé, avec un minimum de faux positifs et une latence d'un chiffre en millisecondes.



Autres cas d’utilisation

- Les moteurs de recommandation peuvent trouver des produits ou des contenus similaires aux utilisateurs en représentant les éléments sous forme de vecteurs. Les vecteurs sont créés en analysant les attributs et les modèles. Sur la base des modèles et des attributs des utilisateurs, de nouveaux éléments invisibles peuvent être recommandés aux utilisateurs en trouvant les vecteurs les plus similaires déjà notés positivement et alignés sur l'utilisateur.
- Les moteurs de recherche de documents représentent les documents texte sous forme de vecteurs denses de nombres, capturant le sens sémantique. Au moment de la recherche, le moteur convertit une requête de recherche en vecteur et trouve les documents contenant les vecteurs les plus similaires à la requête en utilisant une recherche approximative du voisin le plus proche. Cette approche de similarité vectorielle permet de faire correspondre les documents en fonction de leur signification plutôt que de simplement faire correspondre des mots clés.

Caractéristiques et limites de la recherche vectorielle

Disponibilité de la recherche vectorielle

La configuration MemoryDB activée par la recherche vectorielle est prise en charge sur les types de nœuds R6g, R7g et T4g et est disponible dans toutes les régions où MemoryDB est disponible. AWS

Les clusters existants ne peuvent pas être modifiés pour permettre la recherche. Toutefois, les clusters activés pour la recherche peuvent être créés à partir de clichés de clusters dont la recherche est désactivée.

Restrictions paramétriques

Le tableau suivant indique les limites applicables aux différents éléments de recherche vectorielle :

Élément	Valeur maximale
Nombre de dimensions dans un vecteur	32768
Nombre d'index pouvant être créés	10
Nombre de champs dans un index	50

Élément	Valeur maximale
PIEDS. SEARCHet FT. AGGREGATE TIMEOUTclause (millisecondes)	10 000
Nombre d'étages du pipeline en pieds AGGREGATEcommande	32
Nombre de champs en pieds AGGREGATE LOADclause	1 024
Nombre de champs en pieds AGGREGATE GROUPBYclause	16
Nombre de champs en pieds AGGREGATE SORTBYclause	16
Nombre de paramètres dans FT. AGGREGATE PARAMclause	32
HNSWParamètre M	512
HNSWParamètre EF_ CONSTRUCTION	4096
HNSWParamètre EF_ RUNTIME	4096

Limites d'échelle

La recherche vectorielle pour MemoryDB est actuellement limitée à une seule partition et la mise à l'échelle horizontale n'est pas prise en charge. La recherche vectorielle prend en charge le dimensionnement vertical et le dimensionnement des répliques.

Restrictions opérationnelles

Persistance de l'indice et remblayage

La fonction de recherche vectorielle conserve la définition des index et le contenu de l'index. Cela signifie que lors de toute demande ou événement opérationnel entraînant le démarrage ou le redémarrage d'un nœud, la définition et le contenu de l'index sont restaurés à partir du dernier

instantané et toutes les transactions en attente sont relues à partir du Journal. Aucune action de l'utilisateur n'est requise pour lancer cette opération. La reconstruction est effectuée sous forme d'opération de remblayage dès que les données sont restaurées. Cela équivaut fonctionnellement à l'exécution automatique d'un FT par le système. CREATEcommande pour chaque index défini. Notez que le nœud devient disponible pour les opérations des applications dès que les données sont restaurées, mais probablement avant la fin du remplissage des index, ce qui signifie que les remplissages redeviendront visibles pour les applications. Par exemple, les commandes de recherche utilisant des index de remblayage peuvent être rejetées. Pour plus d'informations sur le remblayage, voirAperçu de la recherche vectorielle.

L'achèvement du remplissage de l'index n'est pas synchronisé entre un index principal et un réplica. Ce manque de synchronisation peut devenir visible de manière inattendue pour les applications. Il est donc recommandé aux applications de vérifier que le remblayage est terminé sur les primaires et sur toutes les répliques avant de lancer des opérations de recherche.

Importation/exportation de snapshots et migration en direct

La présence d'index de recherche dans un RDB fichier limite la transportabilité compatible de ces données. Le format des index vectoriels défini par la fonctionnalité de recherche vectorielle de MemoryDB n'est compris que par un autre cluster activé par les vecteurs de MemoryDB. De plus, les RDB fichiers des clusters de prévisualisation peuvent être importés par la version GA des clusters MemoryDB, qui reconstruira le contenu de l'index lors du chargement du RDB fichier.

Toutefois, RDB les fichiers qui ne contiennent pas d'index ne sont pas restreints de cette manière. Ainsi, les données d'un cluster de prévisualisation peuvent être exportées vers des clusters non prévisualisés en supprimant les index avant l'exportation.

Consommation de mémoire

La consommation de mémoire est basée sur le nombre de vecteurs, le nombre de dimensions, la valeur M et la quantité de données non vectorielles, telles que les métadonnées associées au vecteur ou d'autres données stockées dans l'instance.

La mémoire totale requise est une combinaison de l'espace nécessaire pour les données vectorielles réelles et de l'espace requis pour les indices vectoriels. L'espace requis pour les données vectorielles est calculé en mesurant la capacité réelle requise pour stocker des vecteurs au sein HASH de structures de JSON données et le surdébit par rapport aux dalles de mémoire les plus proches, pour des allocations de mémoire optimales. Chacun des index vectoriels utilise des références

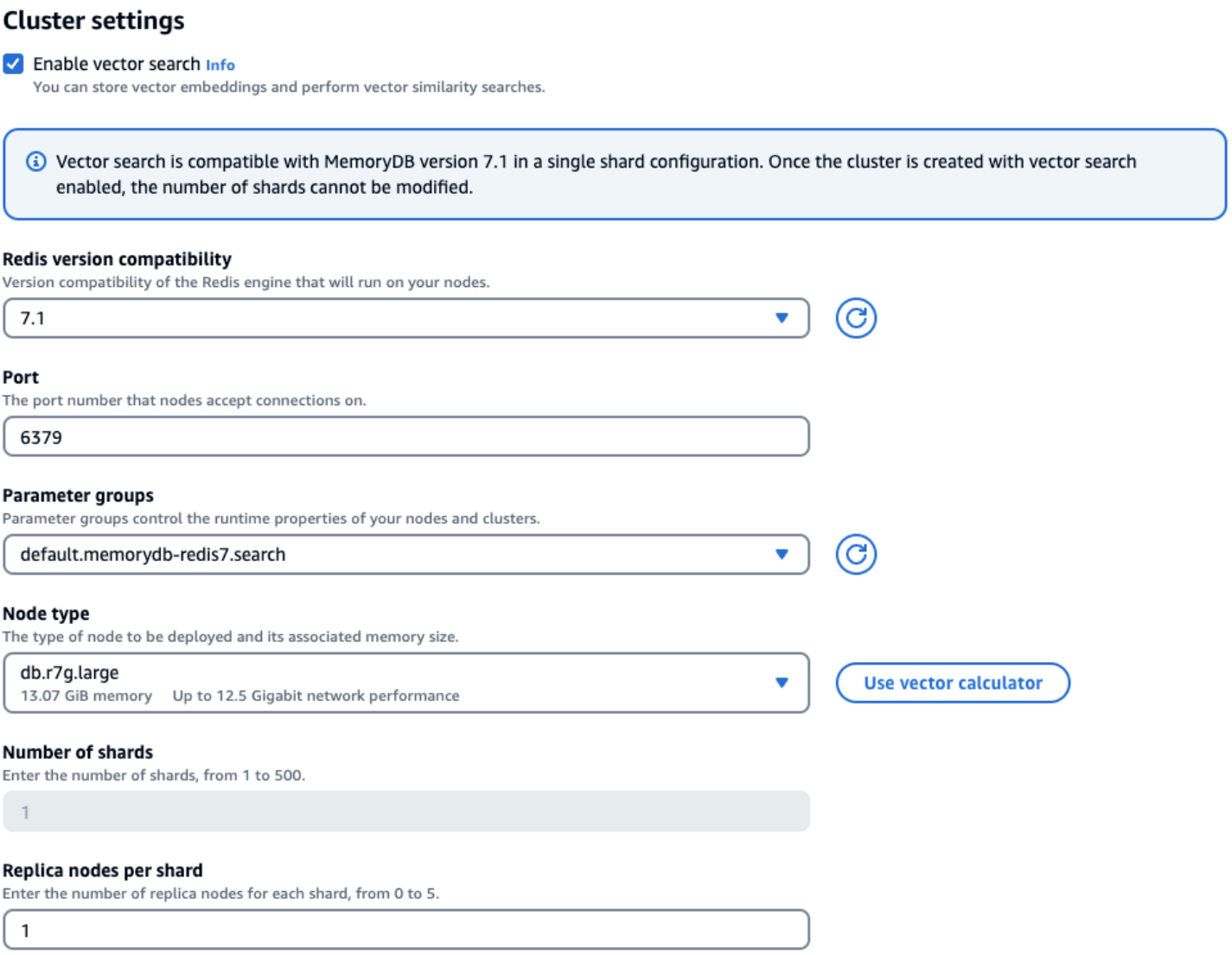
aux données vectorielles stockées dans ces structures de données et utilise des optimisations de mémoire efficaces pour supprimer toute copie dupliquée des données vectorielles dans l'index.

Le nombre de vecteurs dépend de la façon dont vous décidez de représenter vos données sous forme de vecteurs. Par exemple, vous pouvez choisir de représenter un seul document en plusieurs parties, chaque partie représentant un vecteur. Vous pouvez également choisir de représenter l'ensemble du document sous la forme d'un vecteur unique.

Le nombre de dimensions de vos vecteurs dépend du modèle d'intégration que vous choisissez. Par exemple, si vous choisissez d'utiliser le modèle d'intégration AWS Titan, le nombre de dimensions sera de 1536.

Le paramètre M représente le nombre de liens bidirectionnels créés pour chaque nouvel élément lors de la construction de l'index. MemoryDB définit cette valeur par défaut sur 16 ; vous pouvez toutefois la remplacer. Un paramètre M plus élevé fonctionne mieux pour des exigences de dimensionnalité élevées et/ou de rappel élevées, tandis que des paramètres M faibles fonctionnent mieux pour des exigences de faible dimensionnalité et/ou de faible rappel. La valeur M augmente la consommation de mémoire à mesure que l'indice augmente, ce qui augmente la consommation de mémoire.

Dans le cadre de l'expérience console, MemoryDB permet de choisir facilement le type d'instance approprié en fonction des caractéristiques de votre charge de travail vectorielle après avoir coché la case Activer la recherche vectorielle dans les paramètres du cluster.

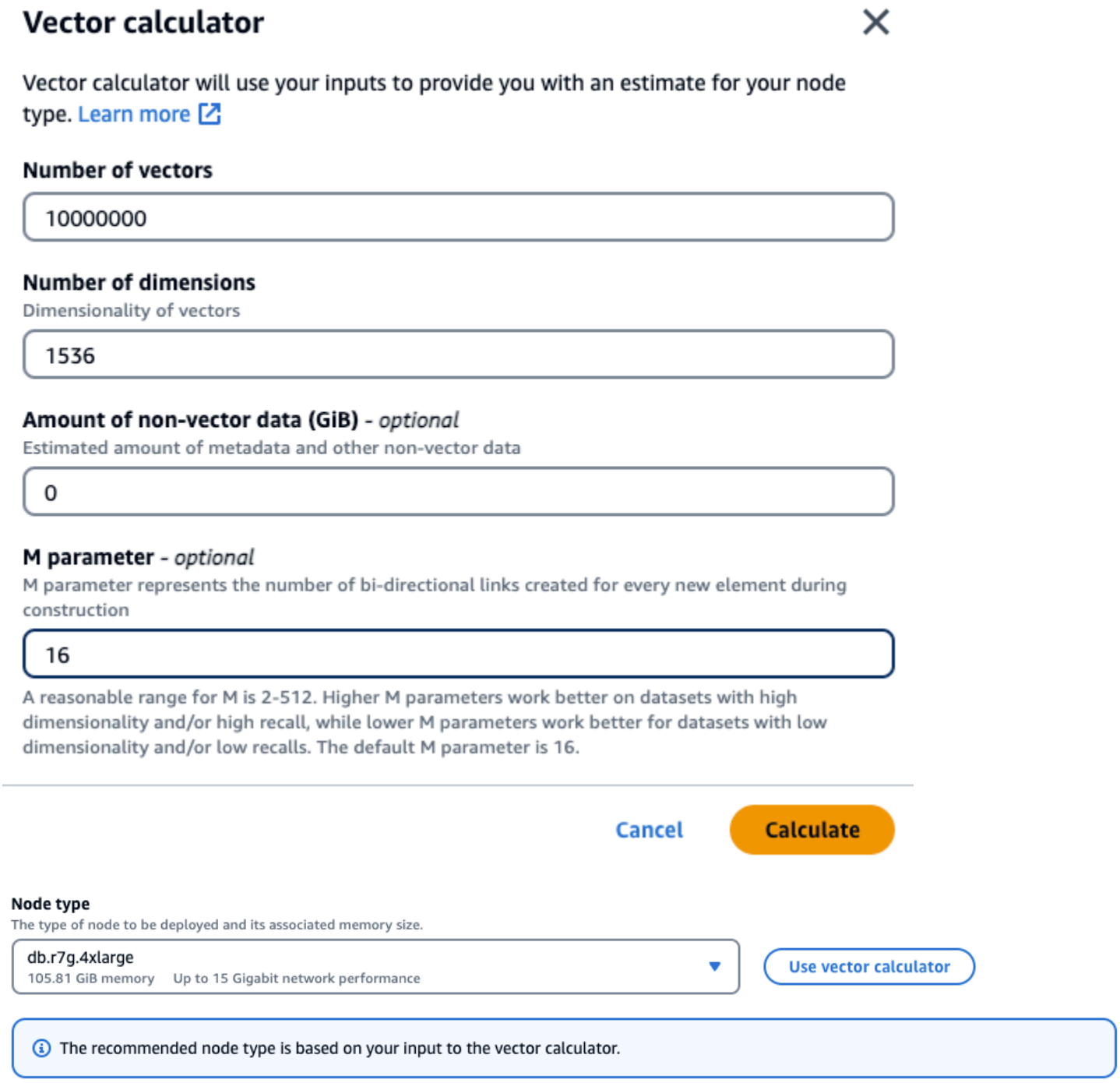


Exemple de charge de travail

Un client souhaite créer un moteur de recherche sémantique basé sur ses documents financiers internes. Ils détiennent actuellement 1 million de documents financiers qui sont découpés en 10 vecteurs par document à l'aide du modèle d'intégration Titan de 1536 dimensions et ne contiennent aucune donnée non vectorielle. Le client décide d'utiliser la valeur par défaut de 16 comme paramètre M.

- Vecteurs : 1 M* 10 morceaux = 10 millions de vecteurs
- Dimensioni : 1536
- Données non vectorielles (Go) : 0 Go
- Paramètre M : 16

Avec ces données, le client peut cliquer sur le bouton Utiliser un calculateur vectoriel dans la console pour obtenir un type d'instance recommandé en fonction de ses paramètres :



Dans cet exemple, le calculateur vectoriel recherchera le plus petit type de nœud MemoryDB r7g capable de contenir la mémoire requise pour stocker les vecteurs en fonction des paramètres fournis. Notez qu'il s'agit d'une approximation et que vous devez tester le type d'instance pour vous assurer qu'il répond à vos besoins.

Sur la base de la méthode de calcul ci-dessus et des paramètres de l'échantillon de charge de travail, ces données vectorielles nécessiteraient 104,9 Go pour stocker les données et un index unique. Dans ce cas, le type d'`db.r7g.4xlarge`instance est recommandé car il dispose de 105,81 Go de mémoire utilisable. Le type de nœud le plus petit suivant serait trop petit pour supporter la charge de travail vectorielle.

Comme chacun des index vectoriels utilise des références aux données vectorielles stockées et ne crée pas de copies supplémentaires des données vectorielles dans l'index vectoriel, les index consommeront également relativement moins d'espace. Cela est très utile pour créer plusieurs index, ainsi que dans les situations où des parties des données vectorielles ont été supprimées et où la reconstruction du HNSW graphe permettrait de créer des connexions de nœuds optimales pour des résultats de recherche vectorielle de haute qualité.

Mémoire insuffisante pendant le remblayage

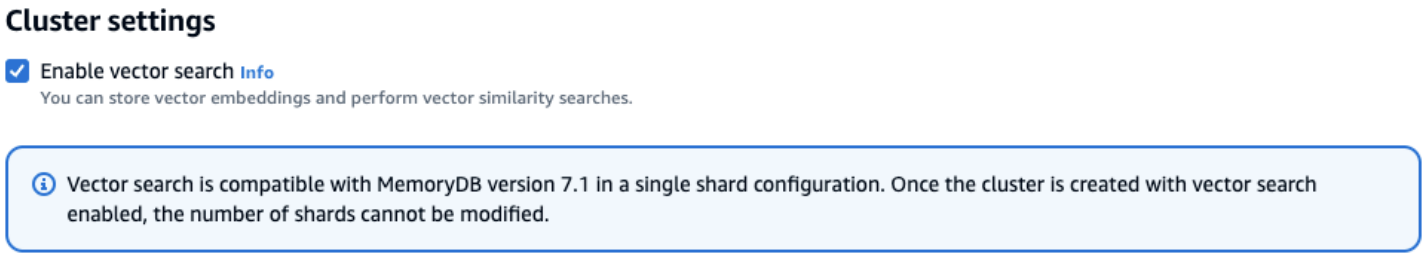
À l'instar des opérations OSS d'écriture Redis, le remplissage d'index est soumis à out-of-memory des limitations. Si OSS la mémoire Redis est pleine alors qu'un remblayage est en cours, tous les remplissages sont interrompus. Si de la mémoire devient disponible, le processus de remblayage est repris. Il est également possible de supprimer et d'indexer lorsque le remplissage est suspendu en raison d'un manque de mémoire.

Transactions

Les commandes`FT.CREATE`,`FT.DROPINDEX`, `FT.ALIASADD`FT.ALIASDEL, et `FT.ALIASUPDATE` ne peuvent pas être exécutées dans un contexte transactionnel, c'est-à-dire pas dans un EXEC blocMULTI/ou dans un FUNCTION script LUA or.

En utilisant le AWS Management Console

Pour créer un cluster activé pour la recherche vectorielle dans la console, vous devez activer la recherche vectorielle dans les paramètres du cluster. La recherche vectorielle est disponible pour la version 7.2 de MemoryDB dans une configuration de partition unique.



Pour plus d'informations sur l'utilisation de la recherche vectorielle avec le AWS Management Console, voirCréation d'un cluster (console).

En utilisant le AWS Command Line Interface

Pour créer un cluster MemoryDB compatible avec la recherche vectorielle, vous pouvez utiliser la commande MemoryDB create-cluster en transmettant un groupe `default.memorydb-redis7.search` de paramètres immuable pour activer les fonctionnalités de recherche vectorielle.

```
aws memorydb create-cluster \
  --cluster-name <value> \
  --node-type <value> \
  --engine redis \
  --engine-version 7.1 \
  --num-shards 1 \
  --acl-name <value> \
  --parameter-group-name default.memorydb-redis7.search
```

Vous pouvez éventuellement créer un nouveau groupe de paramètres pour activer la recherche vectorielle, comme indiqué dans l'exemple suivant. Pour en savoir plus sur les groupes de paramètres, cliquez ici.

```
aws memorydb create-parameter-group \
  --parameter-group-name my-search-parameter-group \
  --family memorydb_redis7
```

Ensuite, mettez à jour le paramètre activé pour la recherche de paramètres sur Oui dans le groupe de paramètres nouvellement créé.

```
aws memorydb update-parameter-group \
  --parameter-group-name my-search-parameter-group \
  --parameter-name-values "ParameterName=search-enabled,ParameterValue=yes"
```

Vous pouvez désormais utiliser ce groupe de paramètres personnalisé au lieu du groupe de paramètres par défaut pour activer la recherche vectorielle sur vos clusters MemoryDB.

Commandes de recherche vectorielle

Vous trouverez ci-dessous une liste des commandes prises en charge pour la recherche vectorielle.

Rubriques

- PIEDS. CREATE
- PIEDS. SEARCH
- PIEDS. AGGREGATE
- PIEDS. DROPINDEX
- PIEDS. INFO
- PIEDS. _ LIST
- PIEDS. ALIASADD
- PIEDS. ALIASDEL
- PIEDS. ALIASUPDATE
- PIEDS. _ ALIASLIST
- PIEDS. PROFILE
- PIEDS. EXPLAIN
- PIEDS. EXPLAINCLI

PIEDS. CREATE

Crée un index et lance un remblayage de cet index. Pour plus d'informations, voir Vue d'ensemble de la recherche vectorielle pour plus de détails sur la construction d'index.

Syntaxe

```
FT.CREATE <index-name>
ON HASH | JSON
[PREFIX <count> <prefix1> [<prefix2>...]]
SCHEMA
(<field-identifier> [AS <alias>]
  NUMERIC
| TAG [SEPARATOR <sep>] [CASESENSITIVE]
| TEXT
| VECTOR [HNSW|FLAT] <attr_count> [<attribute_name> <attribute_value>])

)+
```

Schema (Schéma)

- Identifiant du champ :
  - Pour les clés de hachage, l'identifiant du champ est un nom de champ.
  - Pour JSON les clés, l'identifiant du champ est un JSON chemin.

  Pour plus d'informations, consultez Types de champs d'index.

- Types de champs :
  - TAG: Pour plus d'informations, consultez la section Balises.
  - NUMERIC: Le champ contient un nombre.
  - TEXT: Le champ contient n'importe quel blob de données.
  - VECTOR: champ vectoriel qui prend en charge la recherche vectorielle.
    - Algorithme — Cela peut être HNSW (petit monde navigable hiérarchique) ou FLAT (force brute).
    - `attr_count`— le nombre d'attributs qui seront transmis en tant que configuration de l'algorithme, y compris les noms et les valeurs.
    - `{attribute_name} {attribute_value}`— des paires clé/valeur spécifiques à l'algorithme qui définissent la configuration de l'index.

      Pour l'FLATalgorithme, les attributs sont les suivants :

      Obligatoire :

      - DIM— Nombre de dimensions du vecteur.
      - DISTANCE_ METRIC — Peut être l'un des [L2 | IP |COSINE].
      - TYPE— Type de vecteur. Le seul type pris en charge est`FLOAT32`.

      Facultatif :

      - INITIAL_ CAP — La capacité vectorielle initiale de l'index affecte la taille d'allocation de mémoire de l'index.

      Pour l'HNSWalgorithme, les attributs sont les suivants :

      Obligatoire :

      - TYPE— Type de vecteur. Le seul type pris en charge est`FLOAT32`.
      - DIM— Dimension du vecteur, spécifiée sous la forme d'un entier positif. Maximum : 32 768
      - DISTANCE_ METRIC — Peut être l'un des [L2 | IP |COSINE].

Facultatif :

- INITIAL_ CAP — La capacité vectorielle initiale de l'index affecte la taille d'allocation de mémoire de l'index. La valeur par défaut est 1024.
- M — Nombre maximum d'arêtes sortantes autorisées pour chaque nœud du graphe dans chaque couche. Sur la couche zéro, le nombre maximal d'arêtes sortantes sera de 2 millions. La valeur par défaut est 16. Le maximum est 512.
- EF_ CONSTRUCTION — contrôle le nombre de vecteurs examinés lors de la construction de l'indice. Des valeurs plus élevées pour ce paramètre amélioreront le taux de rappel au détriment de délais de création d'index plus longs. La valeur par défaut est 200. La valeur maximale est 4096.
- EF_ RUNTIME — contrôle le nombre de vecteurs examinés lors des opérations de requête. Des valeurs plus élevées pour ce paramètre peuvent améliorer le rappel au détriment de la durée des requêtes. La valeur de ce paramètre peut être modifiée au cas par cas. Valeur par défaut : 10. La valeur maximale est 4096.

Retour

Renvoie un message OK ou une réponse d'erreur sous forme de chaîne simple.

Exemples

> **Note**
>
> L'exemple suivant utilise des arguments natifs de redis-cli, tels que le déguillement et le déséchappement des données, avant de les envoyer à Redis. OSS Pour utiliser d'autres clients utilisant d'autres langages de programmation (Python, Ruby, C#, etc.), suivez les règles de gestion de ces environnements pour traiter les chaînes et les données binaires. Pour plus d'informations sur les clients pris en charge, voir Outils sur lesquels s'appuyer AWS

Example 1 : Créez des index

Création d'un index pour les vecteurs de taille 2

```
FT.CREATE hash_idx1 ON HASH PREFIX 1 hash: SCHEMA vec AS VEC VECTOR HNSW 6 DIM 2 TYPE
 FLOAT32 DISTANCE_METRIC L2
OK
```

Créez un JSON index en 6 dimensions à l'aide de l'HNSWalgorithme :

```
FT.CREATE json_idx1 ON JSON PREFIX 1 json: SCHEMA $.vec AS VEC VECTOR HNSW 6 DIM 6 TYPE
 FLOAT32 DISTANCE_METRIC L2
OK
```

Example Exemple 2 : renseigner certaines données

Les commandes suivantes sont formatées de manière à pouvoir être exécutées en tant qu'arguments du programme de terminal redis-cli. Les développeurs utilisant des clients utilisant un langage de programmation (tels que Python, Ruby, C#, etc.) devront suivre les règles de gestion de leur environnement pour traiter les chaînes et les données binaires.

Création de certaines données de hachage et de json :

```
HSET hash:0 vec "\x00\x00\x00\x00\x00\x00\x00\x00"
HSET hash:1 vec "\x00\x00\x00\x00\x00\x00\x80\xbf"
JSON.SET json:0 . '{"vec":[1,2,3,4,5,6]}'
JSON.SET json:1 . '{"vec":[10,20,30,40,50,60]}'
JSON.SET json:2 . '{"vec":[1.1,1.2,1.3,1.4,1.5,1.6]}'
```

Notez ce qui suit :

- Les clés du hachage et des JSON données ont les préfixes de leurs définitions d'index.
- Les vecteurs se trouvent aux chemins appropriés des définitions d'index.
- Les vecteurs de hachage sont saisis sous forme de données hexadécimales tandis que les JSON données sont entrées sous forme de nombres.
- Les vecteurs ont les longueurs appropriées, les entrées du vecteur de hachage bidimensionnel contiennent deux flottants de données hexadécimales, les entrées vectorielles json à six dimensions comportent six nombres.

Example Exemple 3 : Supprimer et recréer un index

```
FT.DROPINDEX json_idx1
OK

FT.CREATE json_idx1 ON JSON PREFIX 1 json: SCHEMA $.vec AS VEC VECTOR FLAT 6 DIM 6 TYPE
 FLOAT32 DISTANCE_METRIC L2
```

```
OK
```

Notez que le nouvel JSON index utilise l'`FLAT`algorithme au lieu de l'`HNSW`algorithme. Notez également qu'il réindexera les JSON données existantes :

```
FT.SEARCH json_idx1 "*=>[KNN 100 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 3
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
6) "json:1"
7) 1) "__VEC_score"
   2) "9100"
   3) "$"
   4) "[{\"vec\":[10.0, 20.0, 30.0, 40.0, 50.0, 60.0]}]"
```

PIEDS. SEARCH

Utilise l'expression de requête fournie pour localiser les clés dans un index. Une fois localisés, le nombre et/ou le contenu des champs indexés contenus dans ces clés peuvent être renvoyés. Pour plus d'informations, voir Expression de requête de recherche vectorielle.

Pour créer des données à utiliser dans ces exemples, consultez le FT. CREATEcommande.

Syntaxe

```
FT.SEARCH <index-name> <query>
[RETURN <token_count> (<field-identifier> [AS <alias>])+]
[TIMEOUT timeout]
[PARAMS <count> <name> <value> [<name> <value>]]
[LIMIT <offset> <count>]
[COUNT]
```

- RETURN: Cette clause identifie les champs d'une clé qui sont renvoyés. La clause AS facultative de chaque champ remplace le nom du champ dans le résultat. Seuls les champs déclarés pour cet index peuvent être spécifiés.
- LIMIT: <offset><count>: Cette clause fournit une fonctionnalité de pagination dans la mesure où seules les clés correspondant aux valeurs de décalage et de comptage sont renvoyées. Si cette clause est omise, la valeur par défaut est « LIMIT 0 10 », c'est-à-dire que seul un maximum de 10 clés seront renvoyées.
- PARAMS: deux fois le nombre de paires clé-valeur. Les paires clé/valeur de paramètre peuvent être référencées à partir de l'expression de requête. Pour plus d'informations, voir Expression de requête de recherche vectorielle.
- COUNT: Cette clause supprime le renvoi du contenu des clés, seul le nombre de clés est renvoyé. Il s'agit d'un alias pour « LIMIT 0 0 ».

Retour

Renvoie un tableau ou une réponse d'erreur.

- Si l'opération aboutit, elle renvoie un tableau. Le premier élément est le nombre total de clés correspondant à la requête. Les autres éléments sont des paires de nom de clé et de liste de champs. La liste de champs est un autre tableau comprenant des paires de nom de champ et de valeurs.
- Si l'index est en cours de remplissage à nouveau, la commande renvoie immédiatement une réponse d'erreur.
- Si le délai est dépassé, la commande renvoie une réponse d'erreur.

Exemple : effectuez des recherches

> **Note**
>
> L'exemple suivant utilise des arguments natifs de redis-cli, tels que le déguillement et le déséchappement des données, avant de les envoyer à Redis. OSS Pour utiliser d'autres clients utilisant d'autres langages de programmation (Python, Ruby, C#, etc.), suivez les règles de gestion de ces environnements pour traiter les chaînes et les données binaires. Pour plus d'informations sur les clients pris en charge, voir Outils sur lesquels s'appuyer AWS

Une recherche par hachage

```
FT.SEARCH hash_idx1 "*=>[KNN 2 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00" DIALECT 2
1) (integer) 2
2) "hash:0"
3) 1) "__VEC_score"
   2) "0"
   3) "vec"
   4) "\x00\x00\x00\x00\x00\x00\x00\x00"
4) "hash:1"
5) 1) "__VEC_score"
   2) "1"
   3) "vec"
   4) "\x00\x00\x00\x00\x00\x00\x80\xbf"
```

Cela produit deux résultats, triés en fonction de leur score, qui est la distance par rapport au vecteur de requête (entrée sous forme hexadécimale).

JSONrecherches

```
FT.SEARCH json_idx1 "*=>[KNN 2 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 2
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
```

Cela produit les deux résultats les plus proches, triés en fonction de leur score, et notez que les valeurs JSON vectorielles sont converties en valeurs flottantes et que le vecteur de requête est toujours des données vectorielles. Notez également que le KNN paramètre étant 2, il n'y a que deux résultats. Une valeur plus élevée renverra plus de résultats :

```
FT.SEARCH json_idx1 "*=>[KNN 100 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 3
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
6) "json:1"
7) 1) "__VEC_score"
   2) "9100"
   3) "$"
   4) "[{\"vec\":[10.0, 20.0, 30.0, 40.0, 50.0, 60.0]}]"
```

PIEDS. AGGREGATE

Un surensemble du FT. SEARCHcommande, elle permet un traitement supplémentaire substantiel des clés sélectionnées par l'expression de requête.

Syntaxe

```
FT.AGGREGATE index query
  [LOAD * | [count field [field ...]]]
  [TIMEOUT timeout]
  [PARAMS count name value [name value ...]]
  [FILTER expression]
  [LIMIT offset num]
  [GROUPBY count property [property ...] [REDUCE function count arg [arg ...] [AS name]
 [REDUCE function count arg [arg ...] [AS name] ...]] ...]]
  [SORTBY count [ property ASC | DESC [property ASC | DESC ...]] [MAX num]]
  [APPLY expression AS name]
```

- FILTER, LIMITGROUPBY, SORTBY et les APPLY clauses peuvent être répétées plusieurs fois dans n'importe quel ordre et être librement mélangées. Ils sont appliqués dans l'ordre spécifié, la sortie d'une clause alimentant l'entrée de la clause suivante.

- Dans la syntaxe ci-dessus, une « propriété » est soit un champ déclaré dans le FT. CREATEcommande pour cet index OU la sortie d'une APPLY clause ou d'une REDUCE fonction précédente.
- La LOAD clause est limitée au chargement des champs déclarés dans l'index. « LOAD * » chargera tous les champs déclarés dans l'index.
- Les fonctions de réduction suivantes sont prises en charge : COUNT DISTINCTISHSUM, COUNT _MIN,MAX,AVG,STDDEV,QUANTILE, TOLISTVALUE, FIRST _ et RANDOM _SAMPLE. Pour plus d'informations, voir Agrégations
- LIMIT<offset><count>: Conserve les enregistrements en commençant <offset>et en continuant jusqu'à<count>, tous les autres enregistrements sont supprimés.
- PARAMS: deux fois le nombre de paires clé-valeur. Les paires clé/valeur de paramètre peuvent être référencées à partir de l'expression de requête. Pour plus d'informations, voir Expression de requête de recherche vectorielle.

Retour

Renvoie un tableau ou une réponse d'erreur.

- Si l'opération aboutit, elle renvoie un tableau. Le premier élément est un entier sans signification particulière (doit être ignoré). Les éléments restants sont les résultats produits par la dernière étape. Chaque élément est un tableau de paires de noms de champs et de valeurs.
- Si l'index est en cours de remplissage à nouveau, la commande renvoie immédiatement une réponse d'erreur.
- Si le délai est dépassé, la commande renvoie une réponse d'erreur.

PIEDS. DROPINDEX

Supprime un index. La définition de l'index et le contenu associé sont supprimés. Les OSS touches Redis ne sont pas affectées.

Syntaxe

```
FT.DROPINDEX <index-name>
```

Retour

Renvoie un message OK sous forme de chaîne simple ou une réponse d'erreur.

PIEDS. INFO

Syntaxe

```
FT.INFO <index-name>
```

Sortie du FT. INFOLa page est un tableau de paires clé-valeur, comme décrit dans le tableau suivant :

Clé	Type de la valeur	Description
nom_index	chaîne	Nom de l'index
horodatage de création	entier	Horodatage de style Unix de l'heure de création
type_clé	chaîne	HASHou JSON
key_prefixes	tableau de chaînes	Préfixes clés pour cet index
fields	ensemble d'informations sur le terrain	Champs de cet index
utilisation de l'espace	entier	Octets de mémoire utilisés par cet index
utilisation complète de l'espace	entier	Octets de mémoire utilisés par les champs non vectoriels
utilisation de l'espace vectoriel	entier	Octets de mémoire utilisés par les champs vectoriels
num_docs	entier	Nombre de clés actuellement contenues dans l'index
num_indexed_vector	entier	Nombre de vecteurs actuellement contenus dans l'index
current_lag	entier	Retard d'ingestion récent (milliSeconds)

Clé	Type de la valeur	Description
backfill_status	chaîne	L'un des suivants : terminé InProgres, suspendu ou échoué

Le tableau suivant décrit les informations relatives à chaque champ :

Clé	Type de la valeur	Description
identifiant	chaîne	nom du champ
nom_champ	chaîne	Nom ou JSON chemin du membre de hachage
type	chaîne	l'un des suivants : numérique, balise, texte ou vecteur
option	chaîne	ignore

Si le champ est de type Vector, des informations supplémentaires seront présentes en fonction de l'algorithme.

Pour l'HNSWalgorithme :

Clé	Type de la valeur	Description
automatique	chaîne	HNSW
data_type	chaîne	FLOAT32
métrique de distance	chaîne	l'un des suivants : L2, IP ou Cosine
capacité_initiale	entier	Taille initiale de l'indice du champ vectoriel

Clé	Type de la valeur	Description
capacité_actuelle	entier	Taille actuelle de l'indice du champ vectoriel
arêtes maximales	entier	Paramètre M lors de la création
ef_construction	entier	CONSTRUCTIONParamètre EF_ lors de la création
ef_runtime	entier	RUNTIMEParamètre EF_ lors de la création

Pour l'FLATalgorithme :

Clé	Type de la valeur	Description
automatique	chaîne	FLAT
data_type	chaîne	FLOAT32
métrique de distance	chaîne	l'un des suivants : L2, IP ou Cosine
capacité_initiale	entier	Taille initiale de l'indice du champ vectoriel
capacité_actuelle	entier	Taille actuelle de l'indice du champ vectoriel

PIEDS. _ LIST

Répertoriez tous les index.

Syntaxe

```
FT._LIST
```

Retour

Renvoie un tableau de noms d'index

PIEDS. ALIASADD

Ajoutez un alias pour un index. Le nouveau nom d'alias peut être utilisé partout où un nom d'index est requis.

Syntaxe

```
FT.ALIASADD <alias> <index-name>
```

Retour

Renvoie un message OK sous forme de chaîne simple ou une réponse d'erreur.

PIEDS. ALIASDEL

Supprimez un alias existant pour un index.

Syntaxe

```
FT.ALIASDEL <alias>
```

Retour

Renvoie un message OK sous forme de chaîne simple ou une réponse d'erreur.

PIEDS. ALIASUPDATE

Mettez à jour un alias existant pour qu'il pointe vers un autre index physique. Cette commande n'affecte que les futures références à l'alias. Opérations actuellement en cours (FT. SEARCH, PIEDS. AGGREGATE) ne sont pas affectés par cette commande.

Syntaxe

```
FT.ALIASUPDATE <alias> <index>
```

Retour

Renvoie un message OK sous forme de chaîne simple ou une réponse d'erreur.

PIEDS. _ ALIASLIST

Répertoriez les alias d'index.

Syntaxe

```
FT._ALIASLIST
```

Retour

Renvoie un tableau de la taille du nombre d'alias actuels. Chaque élément du tableau est la paire alias-index.

PIEDS. PROFILE

Exécutez une requête et renvoyez les informations de profil relatives à cette requête.

Syntaxe

```
FT.PROFILE

<index>
SEARCH | AGGREGATE
[LIMITED]
QUERY <query ....>
```

Retour

Un tableau à deux éléments. Le premier élément est le résultat de la `FT.AGGREGATE` commande `FT.SEARCH` or profilée. Le deuxième élément est un ensemble d'informations sur les performances et le profilage.

PIEDS. EXPLAIN

Analyse une requête et renvoie des informations sur la façon dont cette requête a été analysée.

Syntaxe

```
FT.EXPLAIN <index> <query>
```

Retour

Chaîne contenant les résultats analysés.

PIEDS. EXPLAINCLI

Identique au FT. EXPLAINcommande sauf que les résultats sont affichés dans un format différent, plus utile avec le redis-cli.

Syntaxe

```
FT.EXPLAINCLI <index> <query>
```

Retour

Chaîne contenant les résultats analysés.

Sécurité dans MemoryDB

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

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

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. 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 AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à MemoryDB, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de MemoryDB. Il vous montre comment configurer MemoryDB pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources MemoryDB.

Table des matières

- Protection des données dans MemoryDB
- Gestion des identités et des accès dans MemoryDB
- Journalisation et surveillance
- Validation de conformité pour MemoryDB
- Sécurité de l'infrastructure dans MemoryDB
- Confidentialité du trafic inter-réseau
- Mises à jour du service dans MemoryDB

Protection des données dans MemoryDB

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des AWS services que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

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

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- 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 qu'ils contiennent AWS services.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec ou d'autres AWS services utilisateurs de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous entrez dans des

balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Sécurité des données dans MemoryDB

Pour garantir la sécurité de vos données, MemoryDB et Amazon EC2 fournissent des mécanismes de protection contre tout accès non autorisé à vos données sur le serveur.

MemoryDB fournit également des fonctionnalités de chiffrement pour les données des clusters :

- Le chiffrement des données en transit chiffre vos données lorsqu'elles sont déplacées d'un emplacement à un autre, par exemple de nœuds vers un cluster ou entre votre cluster et votre application.
- Le chiffrement au repos chiffre le journal des transactions et vos données sur disque lors des opérations de capture instantanée.

Vous pouvez également l'utiliser Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL) pour contrôler l'accès des utilisateurs à vos clusters.

Rubriques

- Chiffrement au repos dans MemoryDB
- Chiffrement en transit (TLS) dans MemoryDB
- Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL)
- Authentification avec IAM

Chiffrement au repos dans MemoryDB

Pour garantir la sécurité de vos données, MemoryDB et Amazon S3 proposent différentes méthodes pour restreindre l'accès aux données de vos clusters. Pour plus d'informations, consultez MemoryDB et Amazon VPC et Amazon VPC et Gestion des identités et des accès dans MemoryDB.

Le chiffrement au repos de MemoryDB est toujours activé pour renforcer la sécurité des données en chiffrant les données persistantes. Il chiffre les aspects suivants :

- Données du journal des transactions
- Disque pendant les opérations de synchronisation, de capture d'écran et de swap
- Instantanés stockés dans Amazon S3

MemoryDB propose un chiffrement par défaut (géré par le service) au repos, ainsi que la possibilité d'utiliser vos propres clés racine symétriques gérées par le client dans le service de gestion des AWS clés (KMS).

Les données stockées sur des SSD (disques SSD) dans des clusters compatibles avec la hiérarchisation des données sont toujours chiffrées par défaut.

Pour plus d'informations sur le chiffrement en transit, veuillez consulter Chiffrement en transit (TLS) dans MemoryDB.

Rubriques

- Utilisation de clés gérées par le client à partir de AWS KMS
- consultez aussi

Utilisation de clés gérées par le client à partir de AWS KMS

MemoryDB prend en charge les clés racines symétriques gérées par le client (clé KMS) pour le chiffrement au repos. Les clés KMS gérées par le client sont des clés de chiffrement que vous créez, détenez et gérez dans votre AWS compte. Pour plus d'informations, consultez la section Customer Root Keys dans le Guide du développeur du service de gestion des AWS clés. Les clés doivent être créées dans AWS KMS avant de pouvoir être utilisées avec MemoryDB.

Pour savoir comment créer des clés racines AWS KMS, consultez la section Création de clés dans le guide du développeur du service de gestion des AWS clés.

MemoryDB vous permet de vous intégrer à KMS. AWS Pour plus d'informations, veuillez consulter Utilisation d'octrois dans le Guide du développeur AWS Key Management Service. Aucune action du client n'est requise pour activer l'intégration de MemoryDB avec AWS KMS.

La clé de `kms:ViaService` condition limite l'utilisation d'une clé AWS KMS aux demandes provenant de AWS services spécifiques. À utiliser `kms:ViaService` avec MemoryDB, incluez les deux ViaService noms dans la valeur de la clé de condition :. `memorydb.amazon_region.amazonaws.com` Pour plus d'informations, voir kms : ViaService.

Vous pouvez l'utiliser AWS CloudTrailpour suivre les demandes que MemoryDB envoie en votre AWS Key Management Service nom. Tous les appels d'API AWS Key Management Service liés aux clés gérées par le client ont CloudTrail des journaux correspondants. Vous pouvez également voir les autorisations créées par MemoryDB en appelant l'appel d'API ListGrantsKMS.

Une fois qu'un cluster est chiffré à l'aide d'une clé gérée par le client, tous les instantanés du cluster sont chiffrés comme suit :

- Les instantanés quotidiens automatiques sont chiffrés à l'aide de la clé gérée par le client associée au cluster.
- L'instantané final créé lorsque le cluster est supprimé est également chiffré à l'aide de la clé gérée par le client associée au cluster.
- Les instantanés créés manuellement sont chiffrés par défaut pour utiliser la clé KMS associée au cluster. Vous pouvez la remplacer en choisissant une autre clé gérée par le client.
- La copie d'un instantané utilise par défaut la clé gérée par le client associée à l'instantané source. Vous pouvez la remplacer en choisissant une autre clé gérée par le client.

> **Note**
>
> - Les clés gérées par le client ne peuvent pas être utilisées lors de l'exportation d'instantanés vers le compartiment Amazon S3 que vous avez sélectionné. Cependant, tous les instantanés exportés vers Amazon S3 sont chiffrés à l'aide du chiffrement côté serveur. Vous pouvez choisir de copier le fichier instantané sur un nouvel objet S3 et de le chiffrer à l'aide d'une clé KMS gérée par le client, de copier le fichier dans un autre compartiment S3 configuré avec le chiffrement par défaut à l'aide d'une clé KMS ou de modifier une option de chiffrement dans le fichier lui-même.
> - Vous pouvez également utiliser des clés gérées par le client pour chiffrer des instantanés créés manuellement qui n'utilisent pas de clés gérées par le client pour le chiffrement. Avec

cette option, le fichier de capture enregistré dans Amazon S3 est chiffré à l'aide d'une clé KMS, même si les données ne sont pas chiffrées sur le cluster d'origine.

La restauration à partir d'un instantané vous permet de choisir parmi les options de chiffrement disponibles, à l'instar des options de chiffrement disponibles lors de la création d'un nouveau cluster.

- Si vous supprimez la clé ou si vous la désactivez et que vous révoquez les autorisations relatives à la clé que vous avez utilisée pour chiffrer un cluster, celui-ci devient irrécupérable. En d'autres termes, il ne peut pas être modifié ou restauré après une panne matérielle. AWS KMS supprime les clés racines uniquement après une période d'attente d'au moins sept jours. Une fois la clé supprimée, vous pouvez utiliser une autre clé gérée par le client pour créer un instantané à des fins d'archivage.
- La rotation automatique des clés préserve les propriétés de vos clés racines AWS KMS, de sorte que la rotation n'a aucun effet sur votre capacité à accéder à vos données MemoryDB. Les clusters MemoryDB chiffrés ne prennent pas en charge la rotation manuelle des clés, qui implique la création d'une nouvelle clé racine et la mise à jour des références à l'ancienne clé. Pour en savoir plus, consultez Rotating Customer root keys dans le Guide du développeur du service de gestion des AWS clés.
- Le chiffrement d'un cluster MemoryDB à l'aide d'une clé KMS nécessite une autorisation par cluster. Cette subvention est utilisée pendant toute la durée de vie du cluster. En outre, une subvention par instantané est utilisée lors de la création de l'instantané. Cette subvention est retirée une fois le cliché créé.
- Pour plus d'informations sur les autorisations et les limites AWS KMS, consultez la section Quotas du Guide du développeur du service de gestion des AWS clés.

consultez aussi

- Chiffrement en transit (TLS) dans MemoryDB
- MemoryDB et Amazon VPC et Amazon VPC
- Gestion des identités et des accès dans MemoryDB

Chiffrement en transit (TLS) dans MemoryDB

Pour garantir la sécurité de vos données, MemoryDB et Amazon EC2 fournissent des mécanismes de protection contre tout accès non autorisé à vos données sur le serveur. En fournissant une fonctionnalité de chiffrement en transit, MemoryDB vous fournit un outil que vous pouvez utiliser pour protéger vos données lorsqu'elles sont déplacées d'un endroit à un autre. Par exemple, vous pouvez déplacer des données d'un nœud principal vers un nœud de réplication en lecture au sein d'un cluster, ou entre votre cluster et votre application.

Rubriques

- Présentation du chiffrement en transit
- Consultez aussi

Présentation du chiffrement en transit

Le chiffrement en transit de MemoryDB est une fonctionnalité qui renforce la sécurité de vos données aux points les plus vulnérables, c'est-à-dire lorsqu'elles sont en transit d'un endroit à un autre.

Le chiffrement en transit de MemoryDB implémente les fonctionnalités suivantes :

- Connexions cryptées : les connexions au serveur et au client sont cryptées par le protocole TLS (Transport Layer Security).
- Réplication chiffrée : les données transférées entre un nœud primaire et des nœuds en réplica sont chiffrées.
- Authentification du serveur : les clients peuvent authentifier leur connexion au bon serveur.

Depuis le 20/07/2023, TLS 1.2 est la version minimale prise en charge pour les clusters nouveaux et existants. Utilisez ce lien pour en savoir plus sur le protocole TLS 1.2 à l'adresse AWS.

Pour plus d'informations sur la connexion aux clusters MemoryDB, consultez. Connexion aux nœuds MemoryDB à l'aide de redis-cli

Consultez aussi

- Chiffrement au repos dans MemoryDB
- Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL)

- MemoryDB et Amazon VPC et Amazon VPC
- Gestion des identités et des accès dans MemoryDB

Authentification des utilisateurs à l'aide de listes de contrôle d'accès (ACL)

Vous pouvez authentifier les utilisateurs à l'aide de listes de contrôle d'accès (ACL).

Les ACL vous permettent de contrôler l'accès au cluster en regroupant les utilisateurs. Ces listes de contrôle d'accès sont conçues pour organiser l'accès aux clusters.

Avec les ACL, vous créez des utilisateurs et leur attribuez des autorisations spécifiques à l'aide d'une chaîne d'accès, comme décrit dans la section suivante. Vous assignez les utilisateurs à des listes de contrôle d'accès alignées sur un rôle spécifique (administrateurs, ressources humaines) qui sont ensuite déployées sur un ou plusieurs clusters MemoryDB. Vous pouvez ainsi établir des limites de sécurité entre les clients utilisant le ou les mêmes clusters MemoryDB et empêcher les clients d'accéder aux données des autres.

Les ACL sont conçues pour prendre en charge l'introduction de Redis ACL dans Redis OSS 6. Lorsque vous utilisez des ACL avec votre cluster MemoryDB, certaines limites s'appliquent :

- Vous ne pouvez pas spécifier de mots de passe dans une chaîne d'accès. Vous définissez des mots de passe avec CreateUserou par UpdateUserappels.
- Pour les droits d'utilisateur, vous passez on et `off` dans le cadre de la chaîne d'accès. Si aucune des deux n'est spécifiée dans la chaîne d'accès, l'utilisateur est affecté au cluster `off` et ne dispose pas de droits d'accès.
- Vous ne pouvez pas utiliser de commandes interdites. Si vous spécifiez une commande interdite, une exception sera émise. Pour obtenir la liste de ces commandes, consultezCommandes Redis OSS restreintes.
- Vous ne pouvez pas utiliser la commande `reset` dans le cadre d'une chaîne d'accès. Vous spécifiez les mots de passe avec les paramètres de l'API, et MemoryDB gère les mots de passe. Par conséquent, vous ne pouvez pas utiliser `reset` car il supprimerait tous les mots de passe d'un utilisateur.
- Redis OSS 6 introduit la commande ACL LIST. Cette commande renvoie une liste d'utilisateurs ainsi que les règles de liste ACL appliquées à chaque utilisateur. MemoryDB prend en charge la `ACL LIST` commande, mais ne prend pas en charge le hachage des mots de passe comme le fait Redis OSS. Avec MemoryDB, vous pouvez utiliser l'DescribeUsersopération pour obtenir

des informations similaires, y compris les règles contenues dans la chaîne d'accès. Cependant, DescribeUsersne permet pas de récupérer le mot de passe utilisateur.

Les autres commandes en lecture seule prises en charge par MemoryDB incluent ACL WHOAMI, ACL USERS et ACL CAT. MemoryDB ne prend en charge aucune autre commande ACL basée sur l'écriture.

L'utilisation des ACL avec MemoryDB est décrite plus en détail ci-dessous.

Rubriques

- Définition des autorisations à l'aide d'une chaîne d'accès
- Capacités de recherche vectorielle
- Appliquer des ACL à un cluster pour MemoryDB

Définition des autorisations à l'aide d'une chaîne d'accès

Pour spécifier les autorisations d'accès à un cluster MemoryDB, vous créez une chaîne d'accès et vous l'attribuez à un utilisateur en utilisant le AWS CLI ou. AWS Management Console

Les chaînes d'accès sont définies comme une liste de règles délimitées par des espaces qui sont appliquées à l'utilisateur. Elles définissent les commandes qu'un utilisateur peut exécuter et les clés qu'un utilisateur peut utiliser. Pour exécuter une commande, un utilisateur doit avoir accès à la commande en cours d'exécution et à toutes les clés accessibles par la commande. Les règles sont appliquées de gauche à droite de manière cumulative, et une chaîne plus simple peut être utilisée à la place de celle fournie en cas de redondance dans la chaîne fournie.

Pour plus d'informations sur la syntaxe des règles de liste ACL, veuillez consulter Listes ACL.

Dans l'exemple suivant, la chaîne d'accès représente un utilisateur actif ayant accès à toutes les clés et commandes disponibles.

```
on ~* &* +@all
```

La syntaxe de la chaîne d'accès se décompose comme suit :

- on : l'utilisateur est un utilisateur actif.
- ~* : l'accès est accordé à toutes les clés disponibles.

- `&*`— L'accès est donné à toutes les chaînes pubsub.
- `+@all` : l'accès est accordé à toutes les commandes disponibles.

Les paramètres précédents sont les moins restrictifs. Vous pouvez modifier ces paramètres pour les rendre plus sécurisés.

Dans l'exemple suivant, la chaîne d'accès représente un utilisateur dont l'accès est restreint à l'accès en lecture sur les clés commençant par un keyspace « app:: »

```
on ~app::* -@all +@read
```

Vous pouvez affiner ces autorisations en listant les commandes auxquelles l'utilisateur a accès :

`+`*`command1`* : l'accès de l'utilisateur aux commandes est limité à *`command1`*.

`+@category` : l'accès de l'utilisateur est limité à une catégorie de commandes.

Pour plus d'informations sur l'attribution d'une chaîne d'accès à un utilisateur, veuillez consulter Création d'utilisateurs et de listes de contrôle d'accès à l'aide de la console et de la CLI.

Si vous migrez une charge de travail existante vers MemoryDB, vous pouvez récupérer la chaîne d'accès en appelant`ACL LIST`, en excluant l'utilisateur et tout hachage de mot de passe.

Capacités de recherche vectorielle

> **Note**
>
> Cette fonctionnalité est en version préliminaire pour MemoryDB et est sujette à modification.

En Recherche vectorielle effet, toutes les commandes de recherche appartiennent à la `@search` catégorie et aux catégories existantes `@read``@write`, `@fast` et `@slow` sont mises à jour pour inclure les commandes de recherche. Si un utilisateur n'a pas accès à une catégorie, il n'a accès à aucune commande de cette catégorie. Par exemple, si l'utilisateur n'y a pas accès`@search`, il ne peut exécuter aucune commande liée à la recherche.

Le tableau suivant indique le mappage des commandes de recherche vers les catégories appropriées.

Commandes VSS	@read	@write	@fast	@slow
FT.CREATE		Y	Y	
FT.DROPINDEX		Y	Y	
FT.LIST	Y			Y
FT.INFO	Y		Y	
FT.SEARCH	Y			Y
FT.AGGREGATE	Y			Y
FT.PROFILE	Y			Y
FT.ALIASADD		Y	Y	
FT.ALIASDEL		Y	Y	
FT.ALIASUPDATE		Y	Y	
FT._ALIASLIST	Y			Y
FT.EXPLAIN	Y		Y	
FT.EXPLAINCLI	Y		Y	

Commandes VSS	@read	@write	@fast	@slow
`FT.CONFIG`	Y		Y	

Appliquer des ACL à un cluster pour MemoryDB

Pour utiliser les ACL MemoryDB, procédez comme suit :

1. Créez un ou plusieurs utilisateurs.
2. Créez une ACL et ajoutez des utilisateurs à la liste.
3. Assignez l'ACL à un cluster.

Ces étapes sont décrites en détail ci-dessous.

Rubriques

- Création d'utilisateurs et de listes de contrôle d'accès à l'aide de la console et de la CLI
- Gestion des listes de contrôle d'accès à l'aide de la console et de la CLI
- Affectation de listes de contrôle d'accès aux clusters

Création d'utilisateurs et de listes de contrôle d'accès à l'aide de la console et de la CLI

Les informations utilisateur pour les utilisateurs des ACL sont un nom d'utilisateur, et éventuellement un mot de passe et une chaîne d'accès. La chaîne d'accès fournit le niveau d'autorisation relatif aux clés et commandes. Le nom est propre à l'utilisateur et est transmis au moteur.

Assurez-vous que les autorisations utilisateur que vous fournissez correspondent à l'objectif de l'ACL. Par exemple, si vous créez une ACL appeléeAdministrators, la chaîne d'accès de tout utilisateur que vous ajoutez à ce groupe doit être définie de manière à avoir un accès complet aux touches et aux commandes. Pour les utilisateurs d'une `e-commerce` ACL, vous pouvez définir leurs chaînes d'accès en lecture seule.

MemoryDB configure automatiquement un utilisateur par défaut par compte avec un nom d'utilisateur. `"default"` Il ne sera associé à aucun cluster sauf s'il est explicitement ajouté à une ACL. Vous ne pouvez pas le supprimer ou le modifier. Cet utilisateur est destiné à être compatible avec le

comportement par défaut des versions précédentes de Redis OSS et dispose d'une chaîne d'accès qui lui permet d'appeler toutes les commandes et d'accéder à toutes les touches.

Une ACL « open access » immuable sera créée pour chaque compte contenant l'utilisateur par défaut. Il s'agit de la seule ACL dont l'utilisateur par défaut peut être membre. Lorsque vous créez un cluster, vous devez sélectionner une ACL à associer au cluster. Bien que vous ayez la possibilité d'appliquer l'ACL « libre accès » à l'utilisateur par défaut, nous vous recommandons vivement de créer une ACL avec des utilisateurs dont les autorisations sont limitées à leurs besoins commerciaux.

Les clusters sur lesquels le protocole TLS n'est pas activé doivent utiliser l'ACL « open access » pour fournir une authentification ouverte.

Les ACL peuvent être créées sans utilisateur. Une ACL vide n'aurait aucun accès à un cluster et ne pourrait être associée qu'à des clusters compatibles TLS.

Lors de la création d'un utilisateur, vous pouvez configurer jusqu'à deux mots de passe. Lorsque vous modifiez un mot de passe, toutes les connexions existantes aux clusters sont conservées.

Tenez compte en particulier de ces contraintes liées au mot de passe utilisateur lorsque vous utilisez des ACL pour MemoryDB :

- Les mots de passe doivent comporter de 16 à 128 caractères imprimables.
- Les caractères non alphanumériques suivants ne sont pas autorisés : `, "" / @`.

Gestion des utilisateurs avec la console et la CLI

Création d'un utilisateur (console)

Pour créer des utilisateurs sur la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, sélectionnez Utilisateurs.
3. Choisissez Créer un utilisateur
4. Sur la page Créer un utilisateur, entrez un nom.

   Les contraintes d'attribution de noms de cluster sont les suivantes :

   - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.

- Doit commencer par une lettre.
- Ils ne peuvent pas comporter deux traits d'union consécutifs.
- Ils ne peuvent pas se terminer par un trait d'union.

5. Sous Mots de passe, vous pouvez saisir jusqu'à deux mots de passe.
6. Sous Chaîne d'accès, entrez une chaîne d'accès. La chaîne d'accès définit le niveau d'autorisation accordé à l'utilisateur pour les clés et commandes.
7. Pour les tags, vous pouvez éventuellement appliquer des tags pour rechercher et filtrer vos utilisateurs ou suivre vos AWS coûts.
8. Choisissez Créer.

Création d'un utilisateur à l'aide du AWS CLI

Pour créer un utilisateur à l'aide de la CLI

- Utilisez la commande create-user pour créer un utilisateur.

  Pour Linux, macOS ou Unix :

```
aws memorydb create-user \
  --user-name user-name-1 \
  --access-string "~objects:* ~items:* ~public:*" \
  --authentication-mode \
        Passwords="abc",Type=password
```

  Pour Windows :

```
aws memorydb create-user ^
  --user-name user-name-1 ^
  --access-string "~objects:* ~items:* ~public:*" ^
  --authentication-mode \
        Passwords="abc",Type=password
```

Modifier un utilisateur (console)

Pour modifier les utilisateurs sur la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, sélectionnez Utilisateurs.
3. Cliquez sur le bouton radio à côté de l'utilisateur que vous souhaitez modifier, puis choisissez Actions -> Modifier
4. Si vous souhaitez modifier un mot de passe, cliquez sur le bouton radio Modifier les mots de passe. Notez que si vous avez deux mots de passe, vous devez saisir les deux lorsque vous modifiez l'un d'entre eux.
5. Si vous mettez à jour la chaîne d'accès, saisissez-en une nouvelle.
6. Sélectionnez Modifier.

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

Pour modifier un utilisateur à l'aide de la CLI

1. Utilisez la commande update-user pour modifier un utilisateur.
2. Lorsqu'un utilisateur est modifié, les listes de contrôle d'accès associées à l'utilisateur sont mises à jour, ainsi que tous les clusters associés à l'ACL. Toutes les connexions existantes sont maintenues. Voici quelques exemples.

   Pour Linux, macOS ou Unix :

   ```
   aws memorydb update-user \
     --user-name user-name-1 \
     --access-string "~objects:* ~items:* ~public:*"
   ```

   Pour Windows :

   ```
   aws memorydb update-user ^
     --user-name user-name-1 ^
     --access-string "~objects:* ~items:* ~public:*"
   ```

Afficher les détails de l'utilisateur (console)

Pour afficher les détails de l'utilisateur sur la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, sélectionnez Utilisateurs.
3. Choisissez l'utilisateur sous Nom d'utilisateur ou utilisez le champ de recherche pour trouver l'utilisateur.
4. Dans Paramètres utilisateur, vous pouvez consulter la chaîne d'accès, le nombre de mots de passe, le statut et le nom de ressource Amazon (ARN) de l'utilisateur.
5. Sous Listes de contrôle d'accès (ACL), vous pouvez consulter l'ACL à laquelle appartient l'utilisateur.
6. Sous Tags, vous pouvez consulter tous les tags associés à l'utilisateur.

Afficher les détails de l'utilisateur à l'aide du AWS CLI

Utilisez la commande describe-users pour afficher les détails d'un utilisateur.

```
aws memorydb describe-users \
  --user-name my-user-name
```

Supprimer un utilisateur (console)

Pour supprimer des utilisateurs sur la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, sélectionnez Utilisateurs.
3. Cliquez sur le bouton radio à côté de l'utilisateur que vous souhaitez modifier, puis choisissez Actions -> Supprimer
4. Pour confirmer, entrez `delete` dans la zone de texte de confirmation, puis choisissez Supprimer.
5. Pour annuler, choisissez Cancel (Annuler).

Suppression d'un utilisateur à l'aide du AWS CLI

Pour supprimer un utilisateur à l'aide de la CLI

- Utilisez la commande delete-user pour supprimer un utilisateur.

  Le compte est supprimé et retiré de toutes les listes de contrôle d'accès auxquelles il appartient. Voici un exemple.

  Pour Linux, macOS ou Unix :

  ```
  aws memorydb delete-user \
    --user-name user-name-2
  ```

  Pour Windows :

  ```
  aws memorydb delete-user ^
    --user-name user-name-2
  ```

Gestion des listes de contrôle d'accès à l'aide de la console et de la CLI

Vous pouvez créer des listes de contrôle d'accès pour organiser et contrôler l'accès des utilisateurs à un ou plusieurs clusters, comme indiqué ci-dessous.

Utilisez la procédure suivante pour gérer les listes de contrôle d'accès à l'aide de la console.

Création d'une liste de contrôle d'accès (ACL) (console)

Pour créer une liste de contrôle d'accès à l'aide de la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Listes de contrôle d'accès (ACL).
3. Choisissez Create ACL.
4. Sur la page Créer une liste de contrôle d'accès (ACL), entrez un nom d'ACL.

   Les contraintes d'attribution de noms de cluster sont les suivantes :

   - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.

- Doit commencer par une lettre.
- Ils ne peuvent pas comporter deux traits d'union consécutifs.
- Ils ne peuvent pas se terminer par un trait d'union.

5. Sous Utilisateurs sélectionnés, effectuez l'une des opérations suivantes :

   a. Créez un nouvel utilisateur en choisissant Créer un utilisateur

   b. Ajoutez des utilisateurs en choisissant Gérer, puis en sélectionnant des utilisateurs dans la boîte de dialogue Gérer les utilisateurs, puis en sélectionnant Choisir.

6. Pour les balises, vous pouvez éventuellement appliquer des balises pour rechercher et filtrer vos ACL ou suivre vos AWS coûts.

7. Choisissez Créer.

Création d'une liste de contrôle d'accès (ACL) à l'aide du AWS CLI

Utilisez les procédures suivantes pour créer une liste de contrôle d'accès à l'aide de la CLI.

Pour créer une nouvelle ACL et ajouter un utilisateur à l'aide de la CLI

- Utilisez la commande create-acl pour créer une ACL.

  Pour Linux, macOS ou Unix :

```
aws memorydb create-acl \
  --acl-name "new-acl-1" \
  --user-names "user-name-1" "user-name-2"
```

  Pour Windows :

```
aws memorydb create-acl ^
  --acl-name "new-acl-1" ^
  --user-names "user-name-1" "user-name-2"
```

Modifier une liste de contrôle d'accès (ACL) (console)

Pour modifier une liste de contrôle d'accès à l'aide de la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Listes de contrôle d'accès (ACL).
3. Choisissez l'ACL que vous souhaitez modifier, puis choisissez Modifier
4. Sur la page Modifier, sous Utilisateurs sélectionnés, effectuez l'une des opérations suivantes :

   a. Créez un nouvel utilisateur en choisissant Create user à ajouter à l'ACL.

   b. Ajoutez ou supprimez des utilisateurs en choisissant Gérer, puis en sélectionnant ou désélectionnant des utilisateurs dans la boîte de dialogue Gérer les utilisateurs, puis en sélectionnant Choisir.
5. Sur la page Créer une liste de contrôle d'accès (ACL), entrez un nom d'ACL.

   Les contraintes d'attribution de noms de cluster sont les suivantes :

   - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
   - Doit commencer par une lettre.
   - Ils ne peuvent pas comporter deux traits d'union consécutifs.
   - Ils ne peuvent pas se terminer par un trait d'union.
6. Sous Utilisateurs sélectionnés, effectuez l'une des opérations suivantes :

   a. Créez un nouvel utilisateur en choisissant Créer un utilisateur

   b. Ajoutez des utilisateurs en choisissant Gérer, puis en sélectionnant des utilisateurs dans la boîte de dialogue Gérer les utilisateurs, puis en sélectionnant Choisir.
7. Choisissez Modifier pour enregistrer vos modifications ou Annuler pour les ignorer.

Modification d'une liste de contrôle d'accès (ACL) à l'aide du AWS CLI

Pour modifier une ACL en ajoutant de nouveaux utilisateurs ou en supprimant des membres actuels à l'aide de la CLI

- Utilisez la commande update-acl pour modifier une ACL.

  Pour Linux, macOS ou Unix :

```
aws memorydb update-acl --acl-name new-acl-1 \
--user-names-to-add user-name-3 \
--user-names-to-remove user-name-2
```

Pour Windows :

```
aws memorydb update-acl --acl-name new-acl-1 ^
--user-names-to-add user-name-3 ^
--user-names-to-remove user-name-2
```

> **Note**
>
> Cette commande met fin à toutes les connexions ouvertes appartenant à un utilisateur supprimé d'une ACL.

Affichage des détails de la liste de contrôle d'accès (ACL) (console)

Pour afficher les détails de l'ACL sur la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Listes de contrôle d'accès (ACL).
3. Choisissez l'ACL sous le nom de l'ACL ou utilisez le champ de recherche pour trouver l'ACL.
4. Sous Utilisateurs, vous pouvez consulter la liste des utilisateurs associés à l'ACL.
5. Sous Clusters associés, vous pouvez consulter le cluster auquel appartient l'ACL.
6. Sous Tags, vous pouvez consulter tous les tags associés à l'ACL.

Affichage des listes de contrôle d'accès (ACL) à l'aide du AWS CLI

Utilisez la commande describe-acls pour afficher les détails d'une ACL.

```
aws memorydb describe-acls \
  --acl-name test-group
```

Supprimer une liste de contrôle d'accès (ACL) (console)

Pour supprimer des listes de contrôle d'accès à l'aide de la console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Listes de contrôle d'accès (ACL).
3. Choisissez l'ACL que vous souhaitez modifier, puis choisissez Supprimer
4. Sur la page Supprimer, entrez `delete` dans la case de confirmation et choisissez Supprimer ou Annuler pour éviter de supprimer l'ACL.

C'est l'ACL elle-même, et non les utilisateurs appartenant au groupe, qui est supprimée.

Suppression d'une liste de contrôle d'accès (ACL) à l'aide du AWS CLI

Pour supprimer une ACL à l'aide de la CLI

- Utilisez la commande delete-acl pour supprimer une ACL.

  Pour Linux, macOS ou Unix :

  ```
  aws memorydb delete-acl /
     --acl-name
  ```

  Pour Windows :

  ```
  aws memorydb delete-acl ^
     --acl-name
  ```

  Les exemples précédents renvoient la réponse suivante.

  ```
  aws memorydb delete-acl --acl-name "new-acl-1"
  {
      "ACLName": "new-acl-1",
      "Status": "deleting",
      "EngineVersion": "6.2",
      "UserNames": [
          "user-name-1",
          "user-name-3"
      ],
  ```

```
    "clusters": [],
    "ARN":"arn:aws:memorydb:us-east-1:493071037918:acl/new-acl-1"
}
```

Affectation de listes de contrôle d'accès aux clusters

Après avoir créé une ACL et ajouté des utilisateurs, la dernière étape de la mise en œuvre des ACL consiste à attribuer l'ACL à un cluster.

Affectation de listes de contrôle d'accès à des clusters à l'aide de la console

Pour ajouter une ACL à un cluster à l'aide du AWS Management Console, voirCréation d'un cluster MemoryDB.

Affectation de listes de contrôle d'accès à des clusters à l'aide du AWS CLI

L' AWS CLI opération suivante crée un cluster avec le chiffrement en transit (TLS) activé et le acl-name paramètre avec la valeur*my-acl-name*. Remplacez le groupe de sous-réseaux `subnet-group` par un groupe de sous-réseaux existant.

Paramètres clés

- **`--engine-version`**— Doit être 6,2.
- **`--tls-enabled`**— Utilisé pour l'authentification et pour associer une ACL.
- **`--acl-name`**— Cette valeur fournit des listes de contrôle d'accès composées d'utilisateurs dotés d'autorisations d'accès spécifiées pour le cluster.

Pour Linux, macOS ou Unix :

```
aws memorydb create-cluster \
    --cluster-name "new-cluster" \
    --description "new-cluster" \
    --engine-version "6.2" \
    --node-type db.r6g.large \
    --tls-enabled \
    --acl-name "new-acl-1" \
    --subnet-group-name "subnet-group"
```

Pour Windows :

```
aws memorydb create-cluster ^
    --cluster-name "new-cluster" ^
    --cluster-description "new-cluster" ^
    --engine-version "6.2" ^
    --node-type db.r6g.large ^
    --tls-enabled ^
    --acl-name "new-acl-1" ^
    --subnet-group-name "subnet-group"
```

L' AWS CLI opération suivante modifie un cluster dont le chiffrement en transit (TLS) est activé et dont le acl-name paramètre contient la valeur. `new-acl-2`

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name cluster-1 \
    --acl-name "new-acl-2"
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name cluster-1 ^
    --acl-name "new-acl-2"
```

Authentification avec IAM

Rubriques

- Présentation
- Limites
- Configuration
- Connexion

Présentation

Avec l'authentification IAM, vous pouvez authentifier une connexion à MemoryDB à l'aide d'identités AWS IAM, lorsque votre cluster est configuré pour utiliser Redis OSS version 7 ou supérieure.

Cela vous permet de renforcer votre modèle de sécurité et de simplifier de nombreuses tâches administratives de sécurité. Avec l'authentification IAM, vous pouvez configurer un contrôle d'accès précis pour chaque cluster MemoryDB individuel et chaque utilisateur de MemoryDB et suivre les principes d'autorisation du moindre privilège. L'authentification IAM pour MemoryDB fonctionne en fournissant un jeton d'authentification IAM de courte durée au lieu d'un mot de passe utilisateur MemoryDB de longue durée dans le système OSS ou la commande Redis. `AUTH HELLO` Pour plus d'informations sur le jeton d'authentification IAM, reportez-vous au processus de signature Signature version 4 du Guide de référence AWS général et à l'exemple de code ci-dessous.

Vous pouvez utiliser les identités IAM et leurs politiques associées pour restreindre davantage l'accès à Redis OSS. Vous pouvez également accorder l'accès aux utilisateurs depuis leurs fournisseurs d'identité fédérés directement aux clusters MemoryDB.

Pour utiliser AWS IAM avec MemoryDB, vous devez d'abord créer un utilisateur MemoryDB avec le mode d'authentification défini sur IAM, puis vous pouvez créer ou réutiliser une identité IAM. L'identité IAM a besoin d'une politique associée pour accorder l'`memorydb:Connect`action au cluster MemoryDB et à l'utilisateur MemoryDB. Une fois configuré, vous pouvez créer un jeton d'authentification IAM à l'aide des AWS informations d'identification de l'utilisateur ou du rôle IAM. Enfin, vous devez fournir le jeton d'authentification IAM de courte durée sous forme de mot de passe dans votre client Redis OSS lorsque vous vous connectez à votre nœud de cluster MemoryDB. Un client Redis OSS prenant en charge le fournisseur d'informations d'identification peut générer automatiquement les informations d'identification temporaires pour chaque nouvelle connexion. MemoryDB effectuera l'authentification IAM pour les demandes de connexion des utilisateurs de MemoryDB compatibles IAM et validera les demandes de connexion avec IAM.

Limites

Les limites suivantes s'appliquent avec l'authentification IAM :

- L'authentification IAM est disponible lors de l'utilisation du moteur Redis OSS version 7.0 ou supérieure.
- Le jeton d'authentification IAM est valide pendant 15 minutes. Pour les connexions de longue durée, nous vous recommandons d'utiliser un client Redis OSS qui prend en charge une interface de fournisseur d'informations d'identification.
- Une connexion authentifiée IAM à MemoryDB sera automatiquement déconnectée au bout de 12 heures. La connexion peut être prolongée de 12 heures en envoyant une commande `AUTH` ou `HELLO` avec un nouveau jeton d'authentification IAM.
- L'authentification IAM n'est pas prise en charge dans les commandes `MULTI EXEC`.

- Actuellement, l'authentification IAM ne prend pas en charge toutes les clés de contexte de condition globale. Pour plus d'informations sur les clés de contexte de condition globale, consultez Clés de contexte de condition globale AWS dans le Guide de l'utilisateur IAM.

Configuration

Pour configurer une authentification IAM :

1. Créer un cluster

```
aws memorydb create-cluster \
    --cluster-name cluster-01 \
    --description "MemoryDB IAM auth application"
    --node-type db.r6g.large \
    --engine-version 7.0 \
    --acl-name open-access
```

2. Créez un document de stratégie d'approbation IAM, comme indiqué ci-dessous, pour votre rôle afin d'autoriser votre compte à assumer le nouveau rôle. Enregistrez la politique dans un fichier nommé trust-policy.json.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
        "Action": "sts:AssumeRole"
    }
}
```

3. Créez un document de politique IAM, comme indiqué ci-dessous. Enregistrez la politique dans un fichier nommé policy.json.

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

```
      "Resource" : [
        "arn:aws:memorydb:us-east-1:123456789012:cluster/cluster-01",
        "arn:aws:memorydb:us-east-1:123456789012:user/iam-user-01"
      ]
    }
  ]
}
```

4. Créez un rôle IAM.

```
aws iam create-role \
  --role-name "memorydb-iam-auth-app" \
  --assume-role-policy-document file://trust-policy.json
```

5. Créez la politique IAM.

```
aws iam create-policy \
  --policy-name "memorydb-allow-all" \
  --policy-document file://policy.json
```

6. Attachez la politique gérée IAM au rôle.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

7. Créez un compte utilisateur prenant en charge IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
  --authentication-mode Type=iam \
  --access-string "on ~* +@all"
```

8. Créez une ACL et attachez l'utilisateur.

```
aws memorydb create-acl \
  --acl-name iam-acl-01 \
  --user-names iam-user-01

aws memorydb update-cluster \
  --cluster-name cluster-01 \
  --acl-name iam-acl-01
```

Connexion

Se connecter avec un jeton comme mot de passe

Vous devez d'abord générer le jeton d'authentification IAM de courte durée à l'aide d'une Demande pré-signée AWS SigV4. Ensuite, vous fournissez le jeton d'authentification IAM comme mot de passe lors de la connexion à un cluster MemoryDB, comme indiqué dans l'exemple ci-dessous.

```
String userName = "insert user name"
String clusterName = "insert cluster name"
String region = "insert region"

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for MemoryDB.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userName,
 clusterName, region);
String iamAuthToken =
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userName, iamAuthToken)
    .build();

// Create a new Lettuce Redis OSS client
RedisClusterClient client = RedisClusterClient.create(redisURI);
client.connect();
```

Vous trouverez ci-dessous la définition de `IAMAuthTokenRequest`.

```
public class IAMAuthTokenRequest {
    private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
    private static final String REQUEST_PROTOCOL = "http://";
    private static final String PARAM_ACTION = "Action";
```

```
    private static final String PARAM_USER = "User";
    private static final String ACTION_NAME = "connect";
    private static final String SERVICE_NAME = "memorydb";
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final String userName;
    private final String clusterName;
    private final String region;

    public IAMAuthTokenRequest(String userName, String clusterName, String region) {
        this.userName = userName;
        this.clusterName = clusterName;
        this.region = region;
    }

    public String toSignedRequestUri(AWSCredentials credentials) throws
URISyntaxException {
        Request<Void> request = getSignableRequest();
        sign(request, credentials);
        return new URIBuilder(request.getEndpoint())
            .addParameters(toNamedValuePair(request.getParameters()))
            .build()
            .toString()
            .replace(REQUEST_PROTOCOL, "");
    }

    private <T> Request<T> getSignableRequest() {
        Request<T> request  = new DefaultRequest<>(SERVICE_NAME);
        request.setHttpMethod(REQUEST_METHOD);
        request.setEndpoint(getRequestUri());
        request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
        request.addParameters(PARAM_USER, Collections.singletonList(userName));
        return request;
    }

    private URI getRequestUri() {
        return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, clusterName));
    }

    private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
        AWS4Signer signer = new AWS4Signer();
        signer.setRegionName(region);
        signer.setServiceName(SERVICE_NAME);
```

```
        DateTime dateTime = DateTime.now();
        dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));

        signer.presignRequest(request, credentials, dateTime.toDate());
    }

    private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
        return in.entrySet().stream()
            .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
            .collect(Collectors.toList());
    }
}
```

Se connecter avec un fournisseur d'informations d'identification

Le code ci-dessous montre comment s'authentifier auprès de MemoryDB à l'aide du fournisseur d'identifiants d'authentification IAM.

```
String userName = "insert user name"
String clusterName = "insert cluster name"
String region = "insert region"

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
 used as an
// IAM authentication token for MemoryDB.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userName,
 clusterName, region);

// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
    userName, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
```

```
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();

// Create a new Lettuce Redis OSS cluster client
RedisClusterClient client = RedisClusterClient.create(redisURI);
client.connect();
```

Vous trouverez ci-dessous un exemple de client de cluster Lettuce Redis OSS qui intègre l'IAM AuthTokenRequest dans un fournisseur d'informations d'identification pour générer automatiquement des informations d'identification temporaires en cas de besoin.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userName;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userName,
        IAMAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userName;
        this.awsCredentialsProvider = awsCredentialsProvider;
        this.iamAuthTokenRequest = iamAuthTokenRequest;
        this.iamAuthTokenSupplier =
 Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
 TimeUnit.SECONDS);
    }

    @Override
    public Mono<RedisCredentials> resolveCredentials() {
        return Mono.just(RedisCredentials.just(userName, iamAuthTokenSupplier.get()));
    }

    private String getIamAuthToken() {
        return
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
```

Gestion des identités et des accès dans MemoryDB

AWS Identity and Access Management (IAM) est un outil AWS service qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources MemoryDB. IAM est un AWS service outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Comment fonctionne MemoryDB avec IAM
- Exemples de politiques basées sur l'identité pour MemoryDB
- Résolution des problèmes d'identité et d'accès à MemoryDB
- Contrôle d’accès
- Vue d'ensemble de la gestion des autorisations d'accès à vos ressources MemoryDB

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans MemoryDB.

Utilisateur du service : si vous utilisez le service MemoryDB pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités de MemoryDB pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. 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 MemoryDB, consultez. Résolution des problèmes d'identité et d'accès à MemoryDB

Administrateur du service — Si vous êtes responsable des ressources de MemoryDB dans votre entreprise, vous avez probablement un accès complet à MemoryDB. C'est à vous de déterminer les fonctionnalités et les ressources de MemoryDB auxquelles les utilisateurs de votre service doivent

accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec MemoryDB, consultez. Comment fonctionne MemoryDB avec IAM

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez écrire des politiques pour gérer l'accès à MemoryDB. Pour consulter des exemples de politiques basées sur l'identité MemoryDB que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour MemoryDB

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 AWS services 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 AWS services 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 AWS services 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 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 AWS services cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur IAM.
- Accès multiservices — Certains AWS services utilisent des fonctionnalités dans d'autres AWS services. 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 AWS service, associées AWS service 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 AWS services 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 AWS service 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. AWS service Le service peut endosser le rôle afin d’effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier 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. AWS services

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.

Comment fonctionne MemoryDB avec IAM

Avant d'utiliser IAM pour gérer l'accès à MemoryDB, découvrez quelles fonctionnalités IAM peuvent être utilisées avec MemoryDB.

Fonctionnalités IAM que vous pouvez utiliser avec MemoryDB

Fonction IAM	Support de MemoryDB
Politiques basées sur l’identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition d’une politique	Oui
ACL	Oui
ABAC (étiquettes dans les politiques)	Oui
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Fonctions de service	Oui
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble du fonctionnement de MemoryDB et des autres AWS services avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour MemoryDB

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 l'identité pour MemoryDB

Pour consulter des exemples de politiques basées sur l'identité de MemoryDB, consultez. Exemples de politiques basées sur l'identité pour MemoryDB

Politiques basées sur les ressources dans MemoryDB

Prend en charge les politiques basées sur les ressources : Non

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. AWS services

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 la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM.

Actions de politique pour MemoryDB

Soutient les actions politiques : 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 voir la liste des actions MemoryDB, voir Actions définies par MemoryDB dans la référence d'autorisation de service.

Les actions de politique dans MemoryDB utilisent le préfixe suivant avant l'action :

```
MemoryDB
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "MemoryDB:action1",
      "MemoryDB:action2"
         ]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "MemoryDB:Describe*"
```

Pour consulter des exemples de politiques basées sur l'identité de MemoryDB, consultez. Exemples de politiques basées sur l'identité pour MemoryDB

Ressources relatives aux politiques pour MemoryDB

Prend en charge les ressources politiques : 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 voir une liste des types de ressources MemoryDB et de leurs ARN, voir Ressources définies par MemoryDB dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, voir Actions définies par MemoryDB.

Pour consulter des exemples de politiques basées sur l'identité de MemoryDB, consultez. Exemples de politiques basées sur l'identité pour MemoryDB

Clés de conditions de politique pour MemoryDB

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 des exemples de politiques basées sur l'identité de MemoryDB, consultez. Exemples de politiques basées sur l'identité pour MemoryDB

Utilisation de clés de condition

Vous pouvez spécifier des conditions pour déterminer comment une politique IAM prend effet. Dans MemoryDB, vous pouvez utiliser l'`Condition`élément d'une politique JSON pour comparer les clés dans le contexte de la demande avec les valeurs clés que vous spécifiez dans votre politique. Pour plus d'informations, consultez Éléments de politique JSON IAM : condition.

Pour consulter la liste des clés de condition de MemoryDB, voir Clés de condition pour MemoryDB dans la référence d'autorisation de service.

Pour obtenir la liste de toutes les clés de condition globales, veuillez consulter Clés de contexte de condition globales AWS.

Spécification de conditions : Utilisation de clés de condition

Pour mettre en œuvre un contrôle précis, vous pouvez rédiger une politique d'autorisation IAM qui spécifie les conditions permettant de contrôler un ensemble de paramètres individuels pour certaines demandes. Vous pouvez ensuite appliquer la politique aux utilisateurs, groupes ou rôles IAM que vous créez à l'aide de la console IAM.

Pour appliquer une condition, vous ajoutez les informations de condition à la déclaration de politique IAM. Par exemple, pour interdire la création d'un cluster MemoryDB avec le protocole TLS désactivé, vous pouvez spécifier la condition suivante dans votre déclaration de politique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "memorydb:CreateCluster"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "memorydb:TLSEnabled": "false"
        }
      }
    }
  ]
}
```

Pour plus d'informations sur le balisage, consultezMarquer vos ressources MemoryDB.

Pour plus d'informations sur l'utilisation d'opérateurs de condition de politique, veuillez consulter Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions.

Exemples de politique : Utilisation de conditions pour un contrôle de paramètre détaillé

Cette section présente des exemples de politiques pour implémenter un contrôle d'accès précis sur les paramètres MemoryDB répertoriés précédemment.

1. MemoryDB:TlsEnabled — Spécifiez que les clusters seront créés uniquement avec le protocole TLS activé.

```
{
    "Version": "2012-10-17",
    "Statement": [
              {
            "Effect": "Allow",
            "Action": [
                "memorydb:CreateCluster"
            ],
            "Resource": [
                "arn:aws:memorydb:*:*:parametergroup/*",
                "arn:aws:memorydb:*:*:subnetgroup/*",
                "arn:aws:memorydb:*:*:acl/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "memorydb:CreateCluster"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Bool": {
                    "memorydb:TLSEnabled": "true"
                }
            }
        }
    ]
}
```

2. memorydb : UserAuthenticationMode : — Spécifiez que les utilisateurs peuvent être créés avec un mode d'authentification de type spécifique (IAM par exemple).

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

```
                "memorydb:Createuser"
            ],
            "Resource": [
                "arn:aws:memorydb:*:*:user/*"
            ],
            "Condition": {
                "StringEquals": {
                    "memorydb:UserAuthenticationMode": "iam"
                }
            }
        }
    ]
}
```

Dans les cas où vous définissez des politiques basées sur le « refus », il est recommandé d'utiliser l'StringEqualsIgnoreCaseopérateur pour éviter tous les appels avec un type de mode d'authentification utilisateur spécifique, quel que soit le cas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "memorydb:CreateUser"
      ],
      "Resource": "*",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "memorydb:UserAuthenticationMode": "password"
        }
      }
    }
  ]
}
```

Listes de contrôle d'accès (ACL) dans MemoryDB

Supporte les ACL : Oui

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.

Contrôle d'accès basé sur les attributs (ABAC) avec MemoryDB

Supporte l'ABAC (balises 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 MemoryDB

Supporte les informations d'identification temporaires : Oui

Certains AWS services ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui AWS services fonctionnent avec des informations d'identification temporaires, consultez AWS services 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 principales interservices pour MemoryDB

Supporte 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 AWS service, associées AWS service 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 AWS services 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.

Rôles de service pour MemoryDB

Supporte les rôles de service : Oui

Une fonction de service est un rôle IAM qu'un service endosse pour accomplir 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 AWS service dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations pour un rôle de service peut interrompre les fonctionnalités de MemoryDB. Modifiez les rôles de service uniquement lorsque MemoryDB fournit des instructions à cet effet.

Rôles liés à un service pour MemoryDB

Prend en charge les rôles liés aux services : Oui

Un rôle lié à un service est un type de rôle de service lié à un. AWS service Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

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.

Exemples de politiques basées sur l'identité pour MemoryDB

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources MemoryDB. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. 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.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par MemoryDB, y compris le format des ARN pour chacun des types de ressources, voir Actions, ressources et clés de condition pour MemoryDB dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console MemoryDB
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources MemoryDB dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique AWS service, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles.

Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console MemoryDB

Pour accéder à la console MemoryDB, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails des ressources MemoryDB de votre. Compte AWS Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console MemoryDB, attachez également la MemoryDB `ConsoleAccess` ou la politique `ReadOnly` AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur 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 IAM 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:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Résolution des problèmes d'identité et d'accès à MemoryDB

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec MemoryDB et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans MemoryDB
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources MemoryDB

Je ne suis pas autorisé à effectuer une action dans MemoryDB

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 `MemoryDB:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 MemoryDB: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 `MemoryDB:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à MemoryDB.

Certains vous AWS services 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 MemoryDB. 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 MemoryDB

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 MemoryDB prend en charge ces fonctionnalités, consultez. Comment fonctionne MemoryDB 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 connaître la différence entre l'utilisation de rôles et de politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM.

Contrôle d'accès

Vous pouvez disposer d'informations d'identification valides pour authentifier vos demandes, mais vous ne pouvez pas créer de ressources MemoryDB ou y accéder sans autorisation. Par exemple, vous devez disposer des autorisations nécessaires pour créer un cluster MemoryDB.

Les sections suivantes décrivent comment gérer les autorisations pour MemoryDB. Nous vous recommandons de lire d'abord la présentation.

- Vue d'ensemble de la gestion des autorisations d'accès à vos ressources MemoryDB
- Utilisation de politiques basées sur l'identité (politiques IAM) pour MemoryDB

Vue d'ensemble de la gestion des autorisations d'accès à vos ressources MemoryDB

Chaque AWS ressource appartient à un AWS compte, et les autorisations de création ou d'accès à une ressource sont régies par des politiques d'autorisation. Un compte administrateur peut attacher des politiques d'autorisations à des identités IAM (c'est-à-dire des utilisateurs, des groupes et des rôles). En outre, MemoryDB prend également en charge l'attachement de politiques d'autorisation aux ressources.

> **Note**
>
> Un administrateur de compte (ou utilisateur administrateur) est un utilisateur doté des privilèges d'administrateur. Pour plus d'informations, consultez Bonnes pratiques IAM dans le Guide de l'utilisateur IAM.

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) du Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM du Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Rubriques

- Ressources et opérations de MemoryDB
- Présentation de la propriété des ressources

- Gestion de l’accès aux ressources
- Utilisation de politiques basées sur l'identité (politiques IAM) pour MemoryDB
- Autorisations de niveau ressource
- Utilisation de rôles liés à un service pour MemoryDB
- AWS politiques gérées pour MemoryDB
- Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions

Ressources et opérations de MemoryDB

Dans MemoryDB, la ressource principale est un cluster.

Ces ressources ont des noms Amazon Resource Name (ARN) uniques qui leur sont associés, comme cela est illustré ci-dessous.

> **Note**
>
> Pour que les autorisations au niveau des ressources soient efficaces, le nom de la ressource sur la chaîne ARN doit être en minuscules.

Type de ressource	Format ARN
Utilisateur	*arn:aws:memorydb : us-east-1:123456789012 : utilisateur/utilisateur1*
Liste de contrôle d'accès (ACL)	*arn:aws:memorydb : us-east-1:123456789012 : acl/myacl*
Cluster	*arn:aws:memorydb : us-east-1:123456789012 : cluster/my-cluster*
Instantané	*arn:aws:memorydb : us-east-1:123456789012 : snapshot/my-snapshot*

Type de ressource	Format ARN
Groupe de paramètres	*arn:aws:memorydb : us-east-1:123456789012 : groupe de paramètres/* my-parameter-group
Groupe de sous-réseaux	*arn:aws:memorydb : us-east-1:123456789012 : subnetgroup/* my-subnet-group

MemoryDB fournit un ensemble d'opérations permettant de travailler avec les ressources MemoryDB. Pour une liste des opérations disponibles, consultez la section Actions MemoryDB.

Présentation de la propriété des ressources

Le propriétaire d'une ressource est le AWS compte qui a créé la ressource. En d'autres termes, le propriétaire de la ressource est le AWS compte de l'entité principale qui authentifie la demande qui crée la ressource. Une entité principale peut être le compte root, un utilisateur IAM ou un rôle IAM. Les exemples suivants illustrent comment cela fonctionne :

- Supposons que vous utilisiez les informations d'identification du compte root de votre AWS compte pour créer un cluster. Dans ce cas, votre AWS compte est le propriétaire de la ressource. Dans MemoryDB, la ressource est le cluster.
- Supposons que vous créiez un utilisateur IAM dans votre AWS compte et que vous accordiez des autorisations pour créer un cluster à cet utilisateur. Dans ce cas, l'utilisateur peut créer un cluster. Toutefois, votre AWS compte, auquel appartient l'utilisateur, est propriétaire de la ressource du cluster.
- Supposons que vous créiez un rôle IAM dans votre AWS compte avec les autorisations nécessaires pour créer un cluster. Dans ce cas, toute personne capable d'assumer ce rôle peut créer un cluster. Votre AWS compte, auquel appartient le rôle, est propriétaire de la ressource du cluster.

Gestion de l’accès aux ressources

Une politique d'autorisation décrit qui a accès à quoi. La section suivante explique les options disponibles pour créer des politiques d'autorisations.

> **Note**
>
> Cette section décrit l'utilisation d'IAM dans le contexte de MemoryDB. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, consultez Qu'est-ce que IAM ? dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des stratégies IAM, consultez Référence de stratégie AWS IAM dans le Guide de l'utilisateur IAM.

Les politiques attachées à une identité IAM sont appelées des politiques basées sur l'identité (politiques IAM). Les stratégies attachées à une ressource sont appelées stratégies basées sur une ressource.

Rubriques

- Politiques basées sur une identité (politiques IAM)
- Spécification des éléments d'une politique : actions, effets, ressources et mandataires
- Spécification de conditions dans une politique

Politiques basées sur une identité (politiques IAM)

Vous pouvez attacher des politiques à des identités IAM. Par exemple, vous pouvez effectuer les opérations suivantes :

- Attacher une politique d'autorisations à un utilisateur ou à un groupe dans votre compte : un administrateur de compte peut utiliser une politique d'autorisations associée à un utilisateur particulier pour accorder des autorisations. Dans ce cas, les autorisations permettent à cet utilisateur de créer une ressource MemoryDB, telle qu'un cluster, un groupe de paramètres ou un groupe de sécurité.
- Attacher une politique d'autorisations à un rôle (accorder des autorisations entre comptes) : vous pouvez attacher une politique d'autorisation basée sur une identité à un rôle IAM afin d'accorder des autorisations entre comptes. Par exemple, l'administrateur du compte A peut créer un rôle pour accorder des autorisations entre comptes à un autre AWS compte (par exemple, le compte B) ou à un AWS service comme suit :
  1. L'administrateur du Compte A crée un rôle IAM et attache une politique d'autorisation à ce rôle qui accorde des autorisations sur les ressources dans le Compte A.

2. L'administrateur du Compte A attache une politique d'approbation au rôle identifiant le Compte B comme principal pouvant assumer ce rôle.
3. L'administrateur du compte B peut ensuite déléguer les autorisations nécessaires pour assumer le rôle à n'importe quel utilisateur du compte B. Cela permet aux utilisateurs du compte B de créer ou d'accéder aux ressources du compte A. Dans certains cas, vous souhaiterez peut-être accorder à un AWS service des autorisations lui permettant d'assumer le rôle. Pour soutenir cette approche, le principal dans la politique d'approbation peut également être un mandataire du service AWS .

Pour en savoir plus sur l'utilisation d'IAM pour déléguer des autorisations, consultez Gestion des accès dans le Guide de l'utilisateur IAM.

Voici un exemple de politique qui permet à un utilisateur d'effectuer l'`DescribeClusters`action pour votre AWS compte. MemoryDB prend également en charge l'identification de ressources spécifiques à l'aide des ARN des ressources pour les actions d'API. (Cette approche est également appelée autorisations au niveau des ressources.)

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "DescribeClusters",
      "Effect": "Allow",
      "Action": [
         "memorydb:DescribeClusters"],
      "Resource": resource-arn
      }
   ]
}
```

Pour plus d'informations sur l'utilisation de politiques basées sur l'identité avec MemoryDB, consultez. Utilisation de politiques basées sur l'identité (politiques IAM) pour MemoryDB Pour de plus amples informations sur les utilisateurs, les groupes, les rôles et les autorisations, veuillez consulter Identités (utilisateurs, groupes et rôles) dans le Guide de l'utilisateur IAM.

Spécification des éléments d'une politique : actions, effets, ressources et mandataires

Pour chaque ressource MemoryDB (voirRessources et opérations de MemoryDB), le service définit un ensemble d'opérations d'API (voir Actions). Pour accorder des autorisations pour ces opérations d'API, MemoryDB définit un ensemble d'actions que vous pouvez spécifier dans une

politique. Par exemple, pour la ressource de cluster MemoryDB, les actions suivantes sont définies : `CreateClusterDeleteCluster`, et. `DescribeClusters` Une opération d'API peut exiger des autorisations pour plusieurs actions.

Voici les éléments les plus élémentaires d'une politique :

- Ressource : dans une politique, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la politique s'applique. Pour plus d'informations, consultez Ressources et opérations de MemoryDB.
- Action : vous utilisez des mots clés d'action pour identifier les opérations de ressource que vous voulez accorder ou refuser. Par exemple, en fonction de ce qui est spécifié`Effect`, l'`memorydb:CreateCluster`autorisation autorise ou refuse à l'utilisateur l'autorisation d'effectuer l'opération MemoryDB`CreateCluster`.
- Effet – Vous spécifiez l'effet produit lorsque l'utilisateur demande l'action spécifique, qui peut être une autorisation ou un refus. Si vous n'accordez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez également explicitement refuser l'accès à une ressource. Par exemple,vous pouvez le faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une politique différente accorde cet accès.
- Principal : dans les politiques basées sur une identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur une ressource, vous spécifiez l'utilisateur, le compte, le service ou une autre entité qui doit recevoir les autorisations (s'applique uniquement aux politiques basées sur une ressource).

Pour en savoir plus sur la syntaxe des stratégies IAM et pour obtenir des descriptions, consultez Référence de stratégie IAM AWS dans le Guide de l'utilisateur IAM.

Pour un tableau présentant toutes les actions de l'API MemoryDB, consultez. Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions

Spécification de conditions dans une politique

Lorsque vous accordez des autorisations, vous pouvez utiliser le langage des politiques IAM afin de spécifier les conditions définissant à quel moment une politique doit prendre effet. Par exemple, il est possible d'appliquer une politique après seulement une date spécifique. Pour plus d'informations sur la spécification de conditions dans un langage de politique, consultez Condition dans le Guide de l'utilisateur IAM.

Utilisation de politiques basées sur l'identité (politiques IAM) pour MemoryDB

Cette rubrique fournit des exemples de politiques basées sur une identité dans lesquelles un administrateur de compte peut attacher des politiques d'autorisation aux identités IAM (c'est-à-dire aux utilisateurs, groupes et rôles).

> **Important**
>
> Nous vous recommandons de lire d'abord les rubriques qui expliquent les concepts de base et les options de gestion de l'accès aux ressources MemoryDB. Pour plus d'informations, consultez Vue d'ensemble de la gestion des autorisations d'accès à vos ressources MemoryDB.

Les sections de cette rubrique couvrent les sujets suivants :

- Autorisations requises pour utiliser la console MemoryDB
- AWS-politiques gérées (prédéfinies) pour MemoryDB
- Exemples de politiques gérées par le client

Un exemple de politique d'autorisation est exposé ci-dessous.

```
{
   "Version": "2012-10-17",
   "Statement": [{
       "Sid": "AllowClusterPermissions",
       "Effect": "Allow",
       "Action": [
          "memorydb:CreateCluster",
          "memorydb:DescribeClusters",
          "memorydb:UpdateCluster"],
       "Resource": "*"
       },
       {
         "Sid": "AllowUserToPassRole",
         "Effect": "Allow",
         "Action": [ "iam:PassRole" ],
         "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
       }
   ]
```

```
}
```

La politique possède deux énoncés:

- La première instruction accorde des autorisations pour les actions MemoryDB (`memorydb:CreateCluster`,`memorydb:DescribeClusters`, et`memorydb:UpdateCluster`) sur tout cluster appartenant au compte.
- La deuxième instruction accorde des autorisations pour l'action IAM (`iam:PassRole`) sur le nom du rôle IAM spécifié à la fin de la valeur `Resource`.

La politique ne spécifie pas l'élément `Principal` car, dans une politique basée sur une identité, vous ne spécifiez pas le principal qui obtient l'autorisation. Quand vous attachez une politique à un utilisateur, l'utilisateur est le principal implicite. Lorsque vous attachez une politique d'autorisation à un rôle IAM, le principal identifié dans la politique d'approbation de ce rôle obtient les autorisations.

Pour un tableau présentant toutes les actions de l'API MemoryDB et les ressources auxquelles elles s'appliquent, consultez. Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions

Autorisations requises pour utiliser la console MemoryDB

Le tableau de référence des autorisations répertorie les opérations de l'API MemoryDB et indique les autorisations requises pour chaque opération. Pour plus d'informations sur les opérations de l'API MemoryDB, consultez. Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions

Pour utiliser la console MemoryDB, accordez d'abord des autorisations pour des actions supplémentaires, comme indiqué dans la politique d'autorisation suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "MinPermsForMemDBConsole",
        "Effect": "Allow",
        "Action": [
            "memorydb:Describe*",
            "memorydb:List*",
            "ec2:DescribeAvailabilityZones",
            "ec2:DescribeVpcs",
```

```
            "ec2:DescribeAccountAttributes",
            "ec2:DescribeSecurityGroups",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:DescribeAlarms",
            "s3:ListAllMyBuckets",
            "sns:ListTopics",
            "sns:ListSubscriptions" ],
        "Resource": "*"
        }
    ]
}
```

La console MemoryDB a besoin de ces autorisations supplémentaires pour les raisons suivantes :

- Les autorisations pour les actions MemoryDB permettent à la console d'afficher les ressources MemoryDB dans le compte.
- La console a besoin d'autorisations pour les actions `ec2` pour interroger Amazon EC2 afin qu'elle puisse afficher les zones de disponibilité, les VPC, les groupes de sécurité et les attributs de compte.
- Les autorisations relatives aux `cloudwatch` actions permettent à la console de récupérer CloudWatch les métriques et les alarmes Amazon et de les afficher dans la console.
- Les autorisations pour les actions `sns` permettent à la console de récupérer les abonnements et les rubriques Amazon Simple Notification Service (Amazon SNS), et de les afficher dans la console.

Exemples de politiques gérées par le client

Si vous n'utilisez pas de politique par défaut et que vous choisissez d'utiliser une politique gérée personnalisée, vous devez assurer l'un des deux points suivants. Vous devez soit avoir les autorisations d'appeler `iam:createServiceLinkedRole` (pour plus d'informations, veuillez consulter Exemple 4 : Autoriser un utilisateur à appeler l'API IAM CreateServiceLinkedRole). Ou vous auriez dû créer un rôle lié au service MemoryDB.

Combinés aux autorisations minimales nécessaires pour utiliser la console MemoryDB, les exemples de politiques présentés dans cette section accordent des autorisations supplémentaires. Les exemples sont également pertinents pour les AWS SDK et les AWS CLI. Pour plus d'informations sur les autorisations nécessaires pour utiliser la console MemoryDB, consultez. Autorisations requises pour utiliser la console MemoryDB

Pour plus d'informations sur la configuration des utilisateurs et des groupes IAM, veuillez consulter Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

> **Important**
>
> Veillez à toujours tester vos politiques IAM de manière approfondie avant de les utiliser. Certaines actions MemoryDB qui semblent simples peuvent nécessiter d'autres actions pour les prendre en charge lorsque vous utilisez la console MemoryDB. Par exemple, `memorydb:CreateCluster` accorde des autorisations pour créer des clusters MemoryDB. Toutefois, pour effectuer cette opération, la console MemoryDB utilise un certain nombre d'`List`actions `Describe` et pour remplir les listes de consoles.

Exemples

- Exemple 1 : autoriser un utilisateur à accéder en lecture seule aux ressources MemoryDB
- Exemple 2 : autoriser un utilisateur à effectuer des tâches courantes d'administrateur système MemoryDB
- Exemple 3 : Autoriser un utilisateur à accéder à toutes les actions de l'API MemoryDB
- Exemple 4 : Autoriser un utilisateur à appeler l'API IAM CreateServiceLinkedRole

Exemple 1 : autoriser un utilisateur à accéder en lecture seule aux ressources MemoryDB

La politique suivante accorde des autorisations pour les actions MemoryDB qui permettent à un utilisateur de répertorier des ressources. En général, vous attachez ce type de politique d'autorisations à un groupe de gestionnaires.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MemDBUnrestricted",
      "Effect":"Allow",
      "Action": [
          "memorydb:Describe*",
          "memorydb:List*"],
      "Resource":"*"
      }
   ]
```

```
}
```

Exemple 2 : autoriser un utilisateur à effectuer des tâches courantes d'administrateur système MemoryDB

Les tâches courantes des administrateurs système incluent la modification des clusters, des paramètres et des groupes de paramètres. Un administrateur système peut également souhaiter obtenir des informations sur les événements MemoryDB. La politique suivante accorde à un utilisateur l'autorisation d'effectuer des actions MemoryDB pour ces tâches courantes d'administrateur système. Généralement, vous attachez ce type de politique d'autorisations au groupe d'administrateurs système.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MDBAllowSpecific",
      "Effect":"Allow",
      "Action":[
          "memorydb:UpdateCluster",
          "memorydb:DescribeClusters",
          "memorydb:DescribeEvents",
          "memorydb:UpdateParameterGroup",
          "memorydb:DescribeParameterGroups",
          "memorydb:DescribeParameters",
          "memorydb:ResetParameterGroup",],
      "Resource":"*"
      }
   ]
}
```

Exemple 3 : Autoriser un utilisateur à accéder à toutes les actions de l'API MemoryDB

La politique suivante permet à un utilisateur d'accéder à toutes les actions MemoryDB. Nous vous conseillons d'accorder ce type de politique d'autorisations uniquement à un utilisateur administrateur.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MDBAllowAll",
      "Effect":"Allow",
      "Action":[
          "memorydb:*" ],
```

```
      "Resource":"*"
      }
   ]
}
```

Exemple 4 : Autoriser un utilisateur à appeler l'API IAM CreateServiceLinkedRole

La politique suivante permet à un utilisateur d'appeler l'API `CreateServiceLinkedRole` IAM. Nous vous recommandons d'accorder ce type de politique d'autorisations à l'utilisateur qui invoque des opérations mutatives de MemoryDB.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"CreateSLRAllows",
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringLike":{
          "iam:AWS ServiceName":"memorydb.amazonaws.com"
        }
      }
    }
  ]
}
```

Autorisations de niveau ressource

Vous pouvez restreindre la portée des autorisations en spécifiant des ressources dans une politique IAM. De nombreuses actions d' AWS CLI API prennent en charge un type de ressource qui varie en fonction du comportement de l'action. Chaque déclaration de politique IAM accorde une autorisation pour une action effectuée sur une ressource. Lorsque l'action n'agit pas sur une ressource nommée, ou lorsque vous accordez l'autorisation d'effectuer l'action sur toutes les ressources, la valeur de la ressource de la politique est un caractère générique (*). Pour la plupart des actions d'API, vous pouvez limiter les ressources qu'un utilisateur peut modifier en spécifiant l'Amazon Resource Name (ARN) d'une ressource ou un modèle ARN qui correspond à plusieurs ressources. Pour limiter les autorisations par ressource, spécifiez la ressource par son ARN.

Format ARN des ressources MemoryDB

> **Note**
>
> Pour que les autorisations au niveau des ressources soient efficaces, le nom de la ressource sur la chaîne ARN doit être en minuscules.

- *Utilisateur – arn:aws:memorydb : us-east- 1:123456789012:utilisateur/utilisateur1*
- *ACL – arn:aws:memorydb : us-east- 1:123456789012:acl/my-acl*
- *Cluster – arn:aws:memorydb : us-east- 1:123456789012 : cluster/my-cluster*
- *Instantané – arn:aws:memorydb : us-east- 1:123456789012 : snapshot/my-snapshot*
- *Groupe de paramètres – arn:aws:memorydb : us-east-1:123456789012:parametergroup/* my-parameter-group
- *Groupe de sous-réseaux – arn:aws:memorydb : us-east-1:123456789012:subnetgroup/* my-subnet-group

Exemples

- Exemple 1 : accorder à un utilisateur un accès complet à des types de ressources MemoryDB spécifiques
- Exemple 2 : refuser à un utilisateur l'accès à un cluster.

Exemple 1 : accorder à un utilisateur un accès complet à des types de ressources MemoryDB spécifiques

La politique suivante autorise explicitement l'accès `account-id` complet spécifié à toutes les ressources de type groupe de sous-réseaux, groupe de sécurité et cluster.

```
{
        "Sid": "Example1",
        "Effect": "Allow",
        "Action": "memorydb:*",
        "Resource": [
             "arn:aws:memorydb:us-east-1:account-id:subnetgroup/*",
             "arn:aws:memorydb:us-east-1:account-id:securitygroup/*",
```

```
            "arn:aws:memorydb:us-east-1:account-id:cluster/*"
        ]
}
```

Exemple 2 : refuser à un utilisateur l'accès à un cluster.

L'exemple suivant refuse explicitement l'`account-id`accès spécifié à un cluster particulier.

```
{
        "Sid": "Example2",
        "Effect": "Deny",
        "Action": "memorydb:*",
        "Resource": [
                "arn:aws:memorydb:us-east-1:account-id:cluster/name"
        ]
}
```

Utilisation de rôles liés à un service pour MemoryDB

MemoryDB utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à un AWS service, tel que MemoryDB. Les rôles liés au service MemoryDB sont prédéfinis par MemoryDB. Ils comprennent toutes les autorisations requises par le service pour appeler des services AWS au nom de vos clusters.

Un rôle lié à un service facilite la configuration de MemoryDB car il n'est pas nécessaire d'ajouter manuellement les autorisations nécessaires. Les rôles existent déjà dans votre AWS compte, mais ils sont liés à des cas d'utilisation de MemoryDB et disposent d'autorisations prédéfinies. Seul MemoryDB peut assumer ces rôles, et seuls ces rôles peuvent utiliser la politique d'autorisation prédéfinie. Vous pouvez supprimer les rôles uniquement après la suppression préalable de leurs ressources connexes. Cela protège vos ressources MemoryDB car vous ne pouvez pas supprimer par inadvertance les autorisations nécessaires pour accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés aux services, consultez Services AWS fonctionnant avec IAM et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Table des matières

- Autorisations de rôle liées à un service pour MemoryDB
- Création d'un rôle lié à un service (IAM)

- Création d'un rôle lié à un service (console IAM)
- Création d'un rôle lié à un service (CLI IAM)
- Création d'un rôle lié à un service (API IAM)

- Modification de la description d'un rôle lié à un service pour MemoryDB
  - Modification de la description d'un rôle lié à un service (console IAM)
  - Modification de la description d'un rôle lié à un service (CLI IAM)
  - Modification de la description d'un rôle lié à un service (API IAM)
- Supprimer un rôle lié à un service pour MemoryDB
  - Nettoyage d'un rôle lié à un service
  - Suppression d'un rôle lié à un service (console IAM)
  - Suppression d'un rôle lié à un service (CLI IAM)
  - Suppression d'un rôle lié à un service (API IAM)

Autorisations de rôle liées à un service pour MemoryDB

MemoryDB utilise le rôle lié à un service nommé AWSServiceRoleForMemoryDB— Cette politique permet à MemoryDB de gérer les AWS ressources en votre nom selon les besoins de gestion de vos clusters.

La politique d'autorisation des rôles AWSServiceRoleForMemoryDB liés au service permet à MemoryDB d'effectuer les actions suivantes sur les ressources spécifiées :

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateTags"
                ],
                "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                        "ec2:CreateAction": "CreateNetworkInterface"
                    },
                    "ForAllValues:StringEquals": {
                        "aws:TagKeys": [
```

```
                    "AmazonMemoryDBManaged"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface"
        ],
        "Resource": [
            "arn:aws:ec2:*:*:network-interface/*",
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:security-group/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DeleteNetworkInterface",
            "ec2:ModifyNetworkInterfaceAttribute"
        ],
        "Resource": "arn:aws:ec2:*:*:network-interface/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/AmazonMemoryDBManaged": "true"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DeleteNetworkInterface",
            "ec2:ModifyNetworkInterfaceAttribute"
        ],
        "Resource": "arn:aws:ec2:*:*:security-group/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeAvailabilityZones",
            "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/MemoryDB"
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez AWS politique gérée : MemoryDB ServiceRolePolicy.

Pour autoriser une entité IAM à créer des rôles liés à un AWSServiceRoleForMemoryDB service

Ajoutez la déclaration de politique suivante aux autorisations de cette entité IAM :

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "memorydb.amazonaws.com"}}
}
```

Pour autoriser une entité IAM à supprimer des rôles liés à un AWSServiceRoleForMemoryDB service

Ajoutez la déclaration de politique suivante aux autorisations de cette entité IAM :

```
{
    "Effect": "Allow",
```

```
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "memorydb.amazonaws.com"}}
}
```

Vous pouvez également utiliser une politique AWS gérée pour fournir un accès complet à MemoryDB.

Création d'un rôle lié à un service (IAM)

Vous pouvez créer un rôle lié à un service à l'aide de la console IAM, de la CLI ou de l'API.

Création d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM pour créer un rôle lié à un service.

Pour créer un rôle lié à un service (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche de la console IAM, sélectionnez Rôles. Ensuite, choisissez Create new role (Créer un nouveau rôle).
3. Sous Sélectionner un type d’entité de confiance, choisissez AWS Service.
4. Sous Ou sélectionnez un service pour afficher ses cas d'utilisation, choisissez MemoryDB.
5. Sélectionnez Next: Permissions (Étape suivante : autorisations).
6. Sous Policy name (Nom de la politique), notez que la `MemoryDBServiceRolePolicy` est nécessaire pour ce rôle. Choisissez Suivant : balises.
7. Notez que les balises ne sont pas prises en charge pour les rôles liés à un service. Choisissez Next: Review (Suivant : vérifier).
8. (Facultatif) Dans le champ Description du rôle, modifiez la description du nouveau rôle lié à un service.
9. Passez en revue les informations du rôle, puis choisissez Créer un rôle.

Création d'un rôle lié à un service (CLI IAM)

Vous pouvez utiliser les opérations IAM depuis le AWS Command Line Interface pour créer un rôle lié à un service. Ce rôle peut inclure la politique d'approbation et les politiques en ligne dont le service a besoin pour endosser le rôle.

Pour créer un rôle lié à un service (CLI)

Utilisez l'opération suivante :

```
$ aws iam create-service-linked-role --aws-service-name memorydb.amazonaws.com
```

Création d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour créer un rôle lié à un service. Ce rôle peut contenir la politique d'approbation et les politiques en ligne dont le service a besoin pour endosser le rôle.

Pour créer un rôle lié à un service (API)

Utilisez l'appel d'API CreateServiceLinkedRole. Dans la demande, spécifiez un nom de service sous la forme `memorydb.amazonaws.com`.

Modification de la description d'un rôle lié à un service pour MemoryDB

MemoryDB ne vous permet pas de modifier le rôle lié au AWSServiceRoleForMemoryDB service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle 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 volet de navigation de gauche de la console IAM, sélectionnez Rôles.
2. Choisissez le nom du rôle à modifier.
3. A l'extrême droite de Description du rôle, choisissez Edit (Modifier).
4. Saisissez une nouvelle description dans la zone et choisissez Save (Enregistrer).

Modification de la description d'un rôle lié à un service (CLI IAM)

Vous pouvez utiliser les opérations 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 l'opération AWS CLI `get-role` for IAM.

   Example

   ```
   $ aws iam get-role --role-name AWSServiceRoleForMemoryDB
   ```

   Utilisez le nom du rôle, pas l'ARN, pour faire référence aux opérations de la CLI. Par exemple, si un rôle a l'ARN : `arn:aws:iam::123456789012:role/myrole`, faites référence au rôle en tant que **`myrole`**.
2. Pour mettre à jour la description d'un rôle lié à un service, utilisez l'opération AWS CLI for IAM. `update-role-description`

   Pour Linux, macOS ou Unix :

   ```
   $ aws iam update-role-description \
       --role-name AWSServiceRoleForMemoryDB \
       --description "new description"
   ```

   Pour Windows :

   ```
   $ aws iam update-role-description ^
       --role-name AWSServiceRoleForMemoryDB ^
       --description "new 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 courante d'un rôle, utilisez l'opération d'API IAM GetRole.

Example

```
https://iam.amazonaws.com/
   ?Action=GetRole
   &RoleName=AWSServiceRoleForMemoryDB
   &Version=2010-05-08
   &AUTHPARAMS
```

2. Pour mettre à jour la description d'un rôle, utilisez l'opération d'API IAM UpdateRoleDescription.

   Example

```
https://iam.amazonaws.com/
   ?Action=UpdateRoleDescription
   &RoleName=AWSServiceRoleForMemoryDB
   &Version=2010-05-08
   &Description="New description"
```

Supprimer un rôle lié à un service pour MemoryDB

Si vous n’avez plus besoin d’utiliser une fonctionnalité ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n’avez aucune entité inutilisée qui n’est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer.

MemoryDB ne supprime pas le rôle lié au service pour vous.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vérifiez d'abord qu'aucune ressource (cluster) n'est associée au rôle.

Pour vérifier si une session est active pour le rôle lié à un service dans la console IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche de la console IAM, sélectionnez Rôles. Choisissez ensuite le nom (et non la case à cocher) du AWSServiceRoleForMemoryDB rôle.
3. Sur la page Récapitulatif du rôle sélectionné, choisissez l'onglet Access Advisor.

4. Dans l'onglet Access Advisor, consultez l'activité récente pour le rôle lié à un service.

Pour supprimer les ressources MemoryDB qui nécessitent AWSServiceRoleForMemoryDB (console)

- Pour supprimer un cluster, consultez les rubriques suivantes :
  - À l'aide du AWS Management Console
  - À l'aide du AWS CLI
  - Utilisation de l'API MemoryDB

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. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche de la console IAM, sélectionnez Rôles. Cochez ensuite la case en regard du nom du rôle que vous souhaitez supprimer, sans sélectionner le nom ou la ligne.
3. Pour les actions sur les Rôle en haut de la page, sélectionnez Supprimer.
4. Sur la page 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. Si vous souhaitez continuer, sélectionnez Oui, supprimer pour lancer la tâche de suppression du rôle.
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.

Suppression d'un rôle lié à un service (CLI IAM)

Vous pouvez utiliser les opérations IAM depuis le AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (CLI)

1. Si vous ne connaissez pas le nom du rôle lié à un service que vous souhaitez supprimer, saisissez la commande suivante. Cette commande répertorie les rôles et leurs noms Amazon Resource Name (ARN) dans votre compte.

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilisez le nom du rôle, pas l'ARN, pour faire référence aux opérations de la 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. 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. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Saisissez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service.

   ```
   $ aws iam delete-service-linked-role --role-name role-name
   ```

3. 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.

Pour supprimer un rôle lié à un service (API)

1. Pour envoyer une demande de suppression pour un rôle lié à un service, appelez DeleteServiceLinkedRole. Dans la demande, spécifiez le nom d'un rôle.

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. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `DeletionTaskId` de la réponse afin de vérifier l'état de la tâche de suppression.

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.

AWS politiques gérées pour MemoryDB

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, voir les politiques AWS gérées dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique ReadOnlyAccess AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour

obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page politiques gérées par AWS pour les fonctions de tâche dans le Guide de l'utilisateur IAM.

AWS politique gérée : MemoryDB ServiceRolePolicy

Vous ne pouvez pas associer la politique ServiceRolePolicy AWS gérée par MemoryDB aux identités de votre compte. Cette politique fait partie du rôle lié au service AWS MemoryDB. Ce rôle permet au service de gérer les interfaces réseau et les groupes de sécurité de votre compte.

MemoryDB utilise les autorisations définies dans cette politique pour gérer les groupes de sécurité et les interfaces réseau EC2. Cela est nécessaire pour gérer les clusters MemoryDB.

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateTags"
                ],
                "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                        "ec2:CreateAction": "CreateNetworkInterface"
                    },
                    "ForAllValues:StringEquals": {
                        "aws:TagKeys": [
                            "AmazonMemoryDBManaged"
                        ]
                    }
                }
            },
            {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AmazonMemoryDBManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
                "Action": [
                    "cloudwatch:PutMetricData"
                ],
                "Resource": "*",
                "Condition": {
                    "StringEquals": {
                        "cloudwatch:namespace": "AWS/MemoryDB"
                    }
                }
            }
        ]
    }
```

AWS-politiques gérées (prédéfinies) pour MemoryDB

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Les politiques gérées octroient les autorisations requises dans les cas d'utilisation courants et vous évitent d'avoir à réfléchir aux autorisations qui sont requises. Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à MemoryDB :

AmazonMemoryDB ReadOnlyAccess

Vous pouvez associer la politique `AmazonMemoryDBReadOnlyAccess` à vos identités IAM. Cette politique accorde des autorisations administratives qui permettent un accès en lecture seule à toutes les ressources MemoryDB.

AmazonMemoryDB ReadOnlyAccess - Accorde un accès en lecture seule aux ressources MemoryDB.

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

```
}
```

AmazonMemoryDB FullAccess

Vous pouvez associer la politique `AmazonMemoryDBFullAccess` à vos identités IAM. Cette politique accorde des autorisations administratives qui permettent un accès complet à toutes les ressources de MemoryDB.

AmazonMemoryDB FullAccess - Accorde un accès complet aux ressources de MemoryDB.

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Effect": "Allow",
   "Action": "memorydb:*",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": "iam:CreateServiceLinkedRole",
   "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB",
   "Condition": {
    "StringLike": {
     "iam:AWSServiceName": "memorydb.amazonaws.com"
    }
   }
  }
 ]
}
```

Vous pouvez également créer vos propres politiques IAM personnalisées pour autoriser les actions de l'API MemoryDB. Vous pouvez attacher ces politiques personnalisées aux utilisateurs ou groupes IAM qui nécessitent ces autorisations.

Mises à jour de MemoryDB pour les politiques gérées AWS

Consultez les détails des mises à jour des politiques AWS gérées pour MemoryDB depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant

les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique des documents de MemoryDB.

Modification	Description	Date
AmazonMemoryDB FullAccess— Ajouter une politique	MemoryDB a ajouté de nouvelles autorisations pour décrire et répertorier les ressources prises en charge. Ces autorisations sont requises pour que MemoryDB puisse interroger toutes les ressources prises en charge dans un compte.	07/10/2021
AmazonMemoryDB ReadOnlyAccess— Ajouter une politique	MemoryDB a ajouté de nouvelles autorisations pour décrire et répertorier les ressources prises en charge. Ces autorisations sont requises pour que MemoryDB puisse créer des applicati ons basées sur un compte en interrogeant toutes les ressources prises en charge dans un compte.	07/10/2021
MemoryDB a commencé à suivre les modifications	Lancement de service	19/08/2021

Autorisations de l'API MemoryDB : référence aux actions, aux ressources et aux conditions

Lorsque vous configurez le contrôle d'accès et que vous écrivez des politiques d'autorisation à associer à une stratégie IAM (basée sur l'identité ou sur les ressources), utilisez le tableau suivant comme référence. Le tableau répertorie chaque opération de l'API MemoryDB et les actions correspondantes pour lesquelles vous pouvez accorder des autorisations pour effectuer l'action. Vous spécifiez les actions dans le champ `Action` de la politique ainsi qu'une valeur des ressources dans le champ `Resource` de la politique. Sauf indication contraire, la ressource est requise. Certains champs incluent à la fois une ressource obligatoire et des ressources facultatives. Lorsqu'il n'y a pas d'ARN de ressource, la ressource de la politique est un caractère générique (*).

> **Note**
>
> Pour indiquer une action, utilisez le préfixe `memorydb:` suivi du nom de l'opération d'API (par exemple, `memorydb:DescribeClusters`).

Journalisation et surveillance

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de MemoryDB et de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller MemoryDB, signaler un problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre CPU l'utilisation ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d'EC2instances Amazon et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une

solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur Amazon CloudWatch Logs.

- AWS CloudTrailcapture API les appels et les événements connexes effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le Guide de l'utilisateur AWS CloudTrail.

Surveillance de MemoryDB avec Amazon CloudWatch

Vous pouvez surveiller MemoryDB en utilisant CloudWatch, qui collecte les données brutes et les traite en métriques lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Les sections suivantes répertorient les métriques et les dimensions de MemoryDB.

Rubriques

- Métriques au niveau de l'hôte
- Métriques pour MemoryDB
- Quelles métriques dois-je surveiller ?
- Choix des périodes et des statistiques de métriques
- CloudWatch Métriques de surveillance

Métriques au niveau de l'hôte

L'espace de `AWS/MemoryDB` noms inclut les métriques suivantes au niveau de l'hôte pour les nœuds individuels.

Voir aussi

- Métriques pour MemoryDB

Métrique	Description	Unité
`CPUUtilization`	Pourcentage d'CPUutilisation pour l'ensemble de l'hôte. Comme Redis OSS est un système à thread unique, nous vous recommandons de surveiller les `EngineCPUUtilization` métriques pour les nœuds de 4 ou plus. vCPUs	Pourcentage
`FreeableMemory`	Espace mémoire disponible sur l'hôte. Cela est dérivé des buffers et du RAM fait que le système d'exploitation considère comme libérables.	Octets
`NetworkBytesIn`	Nombre d'octets lus par l'hôte à partir du réseau.	Octets
`NetworkBytesOut`	Nombre d'octets envoyés par l'instance sur toutes les interfaces réseau.	Octets
`NetworkPacketsIn`	Nombre de paquets reçus par l'instance sur toutes les interfaces réseau. Cette métrique identifie le volume du trafic entrant en ce qui concerne le nombre de paquets sur une seule instance.	Nombre
`NetworkPacketsOut`	Nombre de paquets envoyés par l'instance sur toutes les interfaces réseau. Cette métrique identifie le volume du trafic sortant en ce qui concerne le nombre de paquets sur une seule instance.	Nombre
`NetworkBandwidthInAllowanceExceeded`	Nombre de paquets formés parce que la bande passante agrégée entrante a dépassé le maximum de l'instance.	Nombre
`NetworkConntrackAllowanceExceeded`	Nombre de paquets formés parce que le suivi des connexions a dépassé le maximum de l'instance et que de nouvelles connexions	Nombre

Métrique	Description	Unité
	n'ont pas pu être établies. Cela peut entraîner une perte de paquets pour le trafic vers ou en provenance de l'instance.	
`NetworkBandwidthOutAllowanceExceeded`	Nombre de paquets formés parce que la bande passante agrégée sortante a dépassé le maximum de l'instance.	Nombre
`NetworkPacketsPerSecondAllowanceExceeded`	Nombre de paquets formés parce que le PPS bidirectionnel a dépassé le maximum de l'instance.	Nombre
`NetworkMaxBytesIn`	Nombre maximal d'octets reçus par seconde par minute.	Octets
`NetworkMaxBytesOut`	Nombre maximal d'octets transmis par seconde par minute.	Octets
`NetworkMaxPacketsIn`	Nombre maximal de paquets reçus par seconde par minute.	Nombre
`NetworkMaxPacketsOut`	Nombre maximal de paquets transmis par seconde par minute.	Nombre
`SwapUsage`	Nombre de permutations utilisées sur l'hôte.	Octets

Métriques pour MemoryDB

L'`AWS/MemoryDB`espace de noms inclut les métriques Redis OSS suivantes.

À l'exception de `ReplicationLag` et`EngineCPUUtilization`, ces métriques sont dérivées de la OSS info commande Redis. Chaque métrique est calculée au niveau du nœud.

Pour une documentation complète de la OSS info commande Redis, consultez http://redis.io/commands/info.

Voir aussi

- Métriques au niveau de l'hôte

Métrique	Description	Unité
`ActiveDefragHits`	Nombre de réallocations de valeur par minute effectuées par le processus de défragmentation actif. Ceci est dérivé des `active_defrag_hits` statistiques de OSSINFORedis.	Nombre
`AuthenticationFailures`	Nombre total de tentatives infructueuses d'authentification auprès de Redis à OSS l'aide de la AUTH commande. Vous pouvez trouver plus d'informations sur les échecs d'authentification individuels à l'aide de la ACLLOGcommande. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
`BytesUsedForMemoryDB`	Le nombre total d'octets alloués par MemoryDB à toutes fins utiles, y compris l'ensemble de données, les tampons, etc.	Octets
	`Dimension: Tier=SSD` pour les clusters utilisant Mise à niveau des données : le nombre total d'octets utilisés parSSD.	Octets
	`Dimension: Tier=Memory` pour les clusters utilisant Mise à niveau des données : le nombre total d'octets utilisés par la mémoire. C'est la valeur des `used_memory` statistiques chez Redis OSS INFO.	Octets
`BytesReadFromDisk`	Nombre total d’octets lus sur le disque par minute. Pris en charge uniquement pour les clusters utilisant Mise à niveau des données.	Octets

Métrique	Description	Unité
`BytesWrittenToDisk`	Nombre total d''octets écrits sur le disque par minute. Pris en charge uniquement pour les clusters utilisant Mise à niveau des données.	Octets
`CommandAuthorizationFailures`	Nombre total de tentatives infructueuses par les utilisateurs d'exécuter des commandes qu'ils n'ont pas l'autorisation d'appeler. Vous pouvez trouver plus d'informations sur les échecs d'authentification individuels à l'aide de la ACLLOGcommande. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
`CurrConnections`	Nombre de connexions client, excluant les connexions des réplicas en lecture. MemoryDB utilise deux à quatre connexions pour surveiller le cluster dans chaque cas. Ceci est dérivé des `connected_clients` statistiques de OSSINFORedis.	Nombre
`CurrItems`	Nombre d'éléments dans le cache. Ceci est dérivé de la OSS `keyspace` statistique Redis, qui fait la somme de toutes les touches de l'espace clavier complet.	Nombre
	`Dimension: Tier=Memory` pour les clusters utilisant Mise à niveau des données. Nombre d'éléments en mémoire.	Nombre
	`Dimension: Tier=SSD` (lecteur à état solide) pour les clusters utilisant Mise à niveau des données. Le nombre d'articles contenus dansSSD.	Nombre

Métrique	Description	Unité
`DatabaseMemoryUsagePercentage`	Pourcentage de la mémoire disponible pour le cluster qui est en cours d'utilisation. Ceci est calculé à l'aide `used_memory/maxmemory` de Redis OSS INFO.	Pourcentage
`DatabaseCapacityUsagePercentage`	Pourcentage de la capacité de données totale pour le cluster en cours d'utilisation. Sur les instances Data Tiered, la métrique est calculée comme`(used_memory - mem_not_counted_for_evict + SSD used) / (maxmemory + SSD total capacity)` , où `used_memory` et provient `maxmemory` de Redis OSS INFO. Dans tous les autres cas, la métrique est calculée à l'aide de`used_memory/maxmemory` .	Pourcentage
`DB0AverageTTL`	Expose `avg_ttl` un DBO extrait des `keyspace` statistiques de la commande Redis OSS INFO.	Millisecondes

Métrique	Description	Unité
`EngineCPUUtilization`	Permet CPU d'utiliser le thread du OSS moteur Redis. OSSRedis étant monothrea d, vous pouvez utiliser cette métrique pour analyser la charge du processus Redis OSS lui-même. La `EngineCPUUtilization` métrique fournit une visibilité plus précise du OSS processus Redis. Vous pouvez l'utilise r conjointement avec la `CPUUtilization` métrique. `CPUUtilization` expose CPU l'utilisation de l'instance de serveur dans son ensemble, y compris les autres systèmes d'exploitation et processus de gestion. Pour les types de nœuds plus importants comptant quatre nœuds vCPUs ou plus, utilisez la `EngineCPUUtilization` métrique pour surveiller et définir des seuils de dimension nement. **Note** Sur un hôte MemoryDB, des processus d'arrière-plan surveillent l'hôte pour fournir une expérience de base de données gérée. Ces processus d'arrière-plan peuvent occuper une part importante de la CPU charge de travail. Cela n'est pas significatif pour les hôtes plus grands qui en ont plus de deuxvCPUs. Mais cela peut affecter les petits hôtes qui en comptent 2 vCPUs ou moins. Si vous ne surveille z que la `EngineCPUUtilization` métrique, vous ne serez pas au courant des situations dans lesquelles l'hôte est surchargé à la fois en raison CPU	Pourcentage

Métrique	Description	Unité
	d'une utilisation élevée de Redis OSS et d'une CPU utilisation élevée due aux processus de surveillance en arrière-p lan. Par conséquent, nous recommand ons de surveiller la `CPUUtilization` métrique pour les hôtes dont le nombre est inférieur vCPUs ou égal à deux.	
`Evictions`	Nombre de clés qui ont été expulsées en raison de la limite `maxmemory` . Ceci est dérivé des `evicted_keys` statistiques de OSSINFORedis.	Nombre
`IsPrimary`	Indique si le nœud est le nœud principal de la partition actuelle. La métrique peut être égale à 0 (non primaire) ou 1 (primaire).	Nombre
`KeyAuthorizationFailures`	Nombre total de tentatives infructueuses par les utilisateurs d'accéder aux clés auxquelles ils n'ont pas l'autorisation d'accéder. Vous pouvez trouver plus d'informations sur les échecs d'authentification individuels à l'aide de la ACLLOGcommande. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
`KeyspaceHits`	Le nombre de recherches réussies de clés en lecture seule dans le dictionnaire principal. Ceci est dérivé des `keyspace_hits` statistiques de OSSINFORedis.	Nombre

Métrique	Description	Unité
KeyspaceMisses	Le nombre de recherches non-réussies de clés en lecture seule dans le dictionnaire principal . Ceci est dérivé des keyspace_misses statistiques de OSSINFORedis.	Nombre
KeysTracked	Le nombre de clés suivies par Redis OSS Key Tracking en pourcentage detracking-table-max-keys . Le suivi des clés est utilisé pour faciliter la mise en cache côté client et avertit les clients lorsque les clés sont modifiées.	Nombre
MaxReplicationThroughput	Débit de réplication maximal observé lors du dernier cycle de mesure.	Octets par seconde
MemoryFragmentationRatio	Indique l'efficacité de l'allocation de mémoire du OSS moteur Redis. Certains seuils indiquent différents comportements. La valeur recommandée est d'avoir une fragmentation supérieure à 1.0. Ceci est calculé à partir mem_fragmentation_ratio statistic de Redis OSS INFO.	Nombre
NewConnections	Nombre total de connexions qui ont été acceptées par le serveur au cours de cette période. Ceci est dérivé des total_connections_received statistiques de OSSINFORedis.	Nombre
NumItemsReadFromDisk	Nombre total d'éléments récupérés à partir du disque par minute. Pris en charge uniquement pour les clusters utilisant Mise à niveau des données.	Nombre

Métrique	Description	Unité
`NumItemsWrittenToDisk`	Nombre total d'éléments écrits sur disque par minute. Pris en charge uniquement pour les clusters utilisant Mise à niveau des données.	Nombre
`PrimaryLinkHealthStatus`	Cet état a deux valeurs : 0 ou 1. La valeur 0 indique que les données du nœud principal de MemoryDB ne sont pas synchronisées avec Redis OSS activé. EC2 Une valeur égale à 1 signifie que les données sont synchronisées.	Booléen
`Reclaimed`	Nombre total d'événements d'expiration de clé. Ceci est dérivé des `expired_keys` statistiques de OSSINFORedis.	Nombre
`ReplicationBytes`	Pour les nœuds dans une configuration répliquée, `ReplicationBytes` indique le nombre d'octets que le principal envoie à toutes ses réplicas. Cette métrique est représentative de la charge d'écriture sur le cluster. Ceci est dérivé des `master_repl_offset` statistiques de OSSINFORedis.	Octets
`ReplicationDelayedWriteCommands`	Nombre de commandes d'écriture retardées en raison de la réplication synchrone. La réplication peut être retardée en raison de divers facteurs, tels que l'encombrement du réseau ou le dépassement du débit de réplication maximal.	Nombre
`ReplicationLag`	Cette métrique ne s'applique qu'à un nœud de s'exécutant en tant que réplica en lecture. Elle représente le retard, en secondes, de l'application par le réplica des modifications provenant du nœud principal.	Secondes

Voici des regroupements de certains types de commandes, dérivés de info commandstats. La section commandstats fournit des statistiques basées sur le type de commande, y compris le nombre d'appels.

Pour une liste complète des commandes disponibles, consultez les commandes Redis dans la documentation RedisOSS.

Métrique	Description	Unité
`EvalBasedCmds`	Nombre total de commandes pour les commandes basées sur eval. Ceci est dérivé de la OSS `commandstats` statistique Redis. Ceci est dérivé de la OSS `commandstats` statistique Redis en eval additionnant,. evalsha	Nombre
`GeoSpatialBasedCmds`	Nombre total de commandes pour les commandes basées sur la géolocalisation. Ceci est dérivé de la OSS `commandstats` statistiq ue Redis. Il est dérivé en additionnant tous les types de commandes géo : geoadd, geodist, geohash, geopos, georadius et georadius bymember.	Nombre
`GetTypeCmds`	Le nombre total de commandes basées sur les types de commandes read-only. Ceci est dérivé de la OSS `commandstats` statistique Redis en additionnant toutes les commandes de read-only type (get,hget,scard,lrange, etc.)	Nombre
`HashBasedCmds`	Nombre total de commandes basées sur le hachage. Ceci est dérivé de la OSS `commandstats` statistique Redis en additionn ant toutes les commandes qui agissent sur un ou plusieurs hachages (hget,,,, hkeys hvalshdel, etc.).	Nombre
`HyperLogLogBasedCmds`	Nombre total de commandes basées sur `HyperLogLog` . Ceci est dérivé de la	Nombre

Métrique	Description	Unité
	OSS `commandstats` statistique Redis en additionnant tous les pf types de commandes (pfadd,pfcount,pfmerge, etc.).	
`JsonBasedCmds`	Le nombre total de commandes JSON basées. Ceci est dérivé de la OSS `commandstats` statistique Redis en additionnant toutes les commandes qui agissent sur un ou plusieurs JSON objets du document.	Nombre
`KeyBasedCmds`	Nombre total de commandes basées sur une clé. Ceci est dérivé de la OSS `commandstats` statistique Redis en additionnant toutes les commandes qui agissent sur une ou plusieurs touches dans plusieurs structures de données (del,expire,rename, etc.).	Nombre
`ListBasedCmds`	Nombre total de commandes basées sur une liste. Ceci est dérivé de la OSS `commandstats` statistique Redis en additionnant toutes les commandes qui agissent sur une ou plusieurs listes (lindex,,lrange, lpushltrim, etc.).	Nombre
`PubSubBasedCmds`	Nombre total de commandes pour la fonctionnalité pub/sub. Ceci est dérivé des OSS `commandstats` statistiques Redis en additionnant toutes les commandes utilisées pour les fonctionnalités pub/sub :psubscribe,,, publishpubsub, punsubscribe et. subscribe unsubscribe	Nombre

Métrique	Description	Unité
SearchBasedCmds	Le nombre total de commandes d'index et de recherche secondaires, y compris les commandes de lecture et d'écriture. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant toutes les commandes de recherche qui agissent sur les index secondaires.	Nombre
SearchBasedGetCmds	Nombre total de commandes d'index et de recherche secondaires en lecture seule. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant toutes les commandes d'index secondaire et de recherche.	Nombre
SearchBasedSetCmds	Nombre total de commandes d'écriture et d'index secondaires. Ceci est dérivé de la OSS commandstats statistique Redis en additionn ant toutes les commandes d'index secondaire et d'ensemble de recherche.	Nombre
SearchNumberOfIndexes	Nombre total d'index.	Nombre
SearchNumberOfIndexedKeys	Nombre total de clés Redis OSS indexées	Nombre
SearchTotalIndexSize	Mémoire (octets) utilisée par tous les index.	Octets
SetBasedCmds	Nombre total de commandes basées sur un ensemble. Ceci est dérivé de la OSS commandstats statistique Redis en additionn ant toutes les commandes qui agissent sur un ou plusieurs ensembles (scard,,sdiff, saddsunion, etc.).	Nombre

Métrique	Description	Unité
SetTypeCmds	Le nombre total de commandes de type write. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant tous les mutative types de commandes qui opèrent sur les données (set,hset,sadd,lpop, etc.)	Nombre
SortedSetBasedCmds	Nombre total de commandes qui sont triées en fonction d'un ensemble. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant toutes les commandes qui agissent sur un ou plusieurs ensembles triés (zcount,,,zrange, zrankzadd, etc.).	Nombre
StringBasedCmds	Nombre total de commandes basées sur une chaîne. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant toutes les commandes qui agissent sur une ou plusieurs chaînes (strlen,, setexsetrange, etc.).	Nombre
StreamBasedCmds	Nombre total de commandes basées sur un flux. Ceci est dérivé de la OSS commandstats statistique Redis en additionnant toutes les commandes qui agissent sur un ou plusieurs types de données de flux (xrange,,xlen, xaddxdel, etc.).	Nombre

Quelles métriques dois-je surveiller ?

Les CloudWatch mesures suivantes offrent un bon aperçu des performances de MemoryDB. Dans la plupart des cas, nous vous recommandons de définir des CloudWatch alarmes pour ces mesures afin de pouvoir prendre des mesures correctives avant que des problèmes de performances ne surviennent.

Métriques pour la surveillance

- CPUUtilization
- E ngineCPUUtilization
- SwapUsage
- Evictions
- CurrConnections
- Mémoire
- Réseau
- Réplication

CPUUtilization

Il s'agit d'une métrique au niveau de l'hôte représentée en pourcentage. Pour plus d’informations, consultez Métriques au niveau de l'hôte.

Pour les types de nœuds plus petits avec 2 nœuds vCPUs ou moins, utilisez la `CPUUtilization` métrique pour surveiller votre charge de travail.

D'une manière générale, nous vous suggérons de fixer votre seuil à 90 % de votre disponibilitéCPU. OSSRedis étant monothread, la valeur de seuil réelle doit être calculée en tant que fraction de la capacité totale du nœud. Supposons par exemple que vous utilisiez un type de nœud comportant deux cœurs. Dans ce cas, le seuil CPUUtilization serait de 90/2, soit 45 %. Pour connaître le nombre de cœurs (vCPUs) de votre type de nœud, consultez la section Tarification de MemoryDB.

Vous devrez déterminer votre propre seuil, en fonction du nombre de cœurs du nœud que vous utilisez. Si vous dépassez ce seuil et que votre charge de travail principale provient des demandes de lecture, agrandissez votre cluster en ajoutant des répliques de lecture. Si la charge de travail principale provient de demandes d'écriture, nous vous recommandons d'ajouter des partitions supplémentaires afin de répartir la charge de travail d'écriture sur un plus grand nombre de nœuds principaux.

> **Tip**
>
> Au lieu d'utiliser la métrique au niveau de l'hôte`CPUUtilization`, vous pouvez peut-être utiliser la OSS métrique Redis`EngineCPUUtilization`, qui indique le pourcentage d'utilisation sur le cœur du moteur RedisOSS. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for MemoryDB.

Pour les types de nœuds plus importants avec 4 nœuds vCPUs ou plus, vous pouvez utiliser la `EngineCPUUtilization` métrique, qui indique le pourcentage d'utilisation sur le cœur du OSS moteur Redis. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for MemoryDB.

E ngineCPUUtilization

Pour les types de nœuds plus importants avec 4 nœuds vCPUs ou plus, vous pouvez utiliser la `EngineCPUUtilization` métrique, qui indique le pourcentage d'utilisation sur le cœur du OSS moteur Redis. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for MemoryDB.

SwapUsage

Il s'agit d'une métrique au niveau de l'hôte, publiée en octets. Pour plus d’informations, consultez Métriques au niveau de l'hôte.

Cette métrique ne doit pas dépasser 50 Mo.

Evictions

Il s'agit d'une métrique du moteur. Nous vous recommandons de choisir votre propre seuil d'alarme pour cette métrique en fonction des besoins de votre application.

CurrConnections

Il s'agit d'une métrique du moteur. Nous vous recommandons de choisir votre propre seuil d'alarme pour cette métrique en fonction des besoins de votre application.

Un nombre croissant de CurrConnectionschiffres peut indiquer un problème avec votre application ; vous devrez étudier le comportement de l'application pour résoudre ce problème.

Mémoire

La mémoire est au cœur de RedisOSS. Il est nécessaire de comprendre l'utilisation de la mémoire de votre cluster afin d'éviter la perte de données et de tenir compte de la croissance future de votre jeu de données. Les statistiques relatives à l'utilisation de la mémoire d'un nœud sont disponibles dans la section mémoire de la OSS INFOcommande Redis.

Réseau

L'un des facteurs déterminants de la capacité de bande passante réseau de votre cluster est le type de nœud que vous avez sélectionné. Pour plus d'informations sur la capacité réseau de votre nœud, consultez la tarification d'Amazon MemoryDB.

Réplication

Le volume de données en cours de réplication est visible via le métrique `ReplicationBytes`. Vous pouvez effectuer une surveillance `MaxReplicationThroughput` par rapport au débit de la capacité de réplication. Il est recommandé d'ajouter des partitions supplémentaires lorsque le débit de capacité de réplication maximal est atteint.

`ReplicationDelayedWriteCommands`peut également indiquer si la charge de travail dépasse le débit maximal de capacité de réplication. Pour plus d'informations sur la réplication dans MemoryDB, voir Comprendre la réplication MemoryDB

Choix des périodes et des statistiques de métriques

Bien que CloudWatch vous puissiez choisir n'importe quelle statistique et période pour chaque métrique, toutes les combinaisons ne seront pas utiles. Par exemple, les statistiques moyenne, minimale et maximale pour CPUUtilization sont utiles, mais pas la statistique Sum.

Tous les exemples de MemoryDB sont publiés pendant 60 secondes pour chaque nœud individuel. Pour toute période de 60 secondes, une métrique de nœud ne contiendra qu'un seul échantillon.

CloudWatch Métriques de surveillance

MemoryDB et CloudWatch sont intégrés afin que vous puissiez collecter une variété de métriques. Vous pouvez surveiller ces indicateurs à l'aide de CloudWatch.

> **Note**
>
> Les exemples suivants nécessitent les outils de ligne de CloudWatch commande. Pour plus d'informations sur CloudWatch les outils de développement et pour les télécharger, consultez la page CloudWatch du produit.

Les procédures suivantes vous montrent comment collecter des statistiques CloudWatch d'espace de stockage pour un cluster au cours de la dernière heure.

> **Note**
>
> Les `EndTime` valeurs `StartTime` et fournies dans les exemples suivants sont fournies à titre indicatif. Assurez-vous de remplacer les valeurs d'heure de début et de fin appropriées pour vos nœuds.

Pour plus d'informations sur les limites de MemoryDB, voir Limites de AWS service pour MemoryDB.

CloudWatch Métriques de surveillance (console)

Pour recueillir CPU des statistiques d'utilisation pour un cluster

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse. https://console.aws.amazon.com/memorydb/
2. Sélectionnez les nœuds pour lesquels vous souhaitez consulter les métriques.

> **Note**
>
> L'affichage des métriques sur la console est désactivé si vous sélectionnez plus de 20 nœuds.

a. Sur la page Clusters de la console de AWS gestion, cliquez sur le nom d'un ou de plusieurs clusters.

   La page détaillée du cluster s'affiche.

b. Cliquez sur l'onglet Nodes en haut de la fenêtre.

c. Dans l'onglet Nœuds de la fenêtre détaillée, sélectionnez les nœuds pour lesquels vous souhaitez consulter les métriques.

   La liste des CloudWatch métriques disponibles apparaît en bas de la fenêtre de console.

d. Cliquez sur la métrique CPUd'utilisation.

   La CloudWatch console s'ouvre et affiche les statistiques que vous avez sélectionnées. Vous pouvez utiliser les zones de liste déroulantes Statistic et Period et l'onglet Time Range pour modifier les métriques affichées.

Surveillance des CloudWatch métriques à l'aide du CloudWatch CLI

Pour recueillir CPU des statistiques d'utilisation pour un cluster

- Utilisez la CloudWatch commande aws cloudwatch get-metric-statistics avec les paramètres suivants (notez que les heures de début et de fin ne sont indiquées qu'à titre d'exemple ; vous devrez les remplacer par vos propres heures de début et de fin appropriées) :

  Pour Linux, macOS ou Unix :

```
aws cloudwatch get-metric-statistics CPUUtilization \
    --dimensions=ClusterName=mycluster,NodeId=0002" \
    --statistics=Average \
    --namespace="AWS/MemoryDB" \
    --start-time 2013-07-05T00:00:00 \
    --end-time 2013-07-06T00:00:00 \
    --period=60
```

Pour Windows :

```
mon-get-stats CPUUtilization ^
    --dimensions=ClusterName=mycluster,NodeId=0002" ^
    --statistics=Average ^
    --namespace="AWS/MemoryDB" ^
    --start-time 2013-07-05T00:00:00 ^
    --end-time 2013-07-06T00:00:00 ^
    --period=60
```

Surveillance des CloudWatch métriques à l'aide du CloudWatch API

Pour recueillir CPU des statistiques d'utilisation pour un cluster

- Appelez le CloudWatch API `GetMetricStatistics` avec les paramètres suivants (notez que les heures de début et de fin ne sont indiquées qu'à titre d'exemple ; vous devrez les remplacer par vos propres heures de début et de fin appropriées) :
  - `Statistics.member.1=Average`
  - `Namespace=AWS/MemoryDB`
  - `StartTime=2013-07-05T00:00:00`
  - `EndTime=2013-07-06T00:00:00`
  - `Period=60`
  - `MeasureName=CPUUtilization`
  - `Dimensions=ClusterName=mycluster,NodeId=0002`

  Example

```
http://monitoring.amazonaws.com/
    ?SignatureVersion=4
    &Action=GetMetricStatistics
    &Version=2014-12-01
    &StartTime=2013-07-16T00:00:00
    &EndTime=2013-07-16T00:02:00
    &Period=60
    &Statistics.member.1=Average
```

```
&Dimensions.member.1="ClusterName=mycluster"
&Dimensions.member.2="NodeId=0002"
&Namespace=Amazon/memorydb
&MeasureName=CPUUtilization
&Timestamp=2013-07-07T17%3A48%3A21.746Z
&AWS;AccessKeyId=<&AWS; Access Key ID>
&Signature=<Signature>
```

Surveillance des événements MemoryDB

Lorsque des événements importants se produisent pour un cluster, MemoryDB envoie une notification à un sujet Amazon SNS spécifique. Les exemples incluent des éléments tels que l'échec d'ajout d'un nœud, l'ajout réussi d'un nœud, la modification d'un groupe de sécurité, etc. En surveillant les événements principaux, vous pouvez connaître l'état actuel de vos clusters, et, selon l'événement, prendre des actions correctives.

Rubriques

- Gestion des notifications Amazon de MemoryDB SNS
- Affichage des événements MemoryDB
- Notifications d'événements Amazon SNS

Gestion des notifications Amazon de MemoryDB SNS

Vous pouvez configurer MemoryDB pour envoyer des notifications pour les événements importants du cluster à l'aide d'Amazon Simple Notification Service (AmazonSNS). Dans ces exemples, vous allez configurer un cluster avec le nom de ressource Amazon (ARN) d'un SNS sujet Amazon pour recevoir des notifications.

> **Note**
>
> Cette rubrique part du principe que vous vous êtes inscrit à AmazonSNS, que vous avez créé une SNS rubrique Amazon et que vous vous y êtes abonné. Pour plus d'informations sur Amazon SNS, veuillez consulter le Guide du développeur d'Amazon Simple Notification Service.

Ajouter un SNS sujet Amazon

Les sections suivantes vous montrent comment ajouter un SNS sujet Amazon à l'aide de la AWS console AWS CLI, de ou de MemoryDBAPI.

Ajouter un SNS sujet Amazon (console)

La procédure suivante explique comment ajouter une SNS rubrique Amazon pour un cluster.

> **Note**
>
> Ce processus peut également être utilisé pour modifier le SNS sujet Amazon.

Pour ajouter ou modifier une SNS rubrique Amazon pour un cluster (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse. https://console.aws.amazon.com/memorydb/
2. Dans Clusters, choisissez le cluster pour lequel vous souhaitez ajouter ou modifier un SNS sujet AmazonARN.
3. Sélectionnez Modifier.
4. Dans Modifier le cluster sous Sujet de SNS notification, choisissez le SNS sujet que vous souhaitez ajouter, ou choisissez ARNSaisie manuelle et saisissez le ARN SNS sujet Amazon.
5. Sélectionnez Modifier.

Ajouter un SNS sujet Amazon (AWS CLI)

Pour ajouter ou modifier une SNS rubrique Amazon pour un cluster, utilisez la AWS CLI commande`update-cluster`.

L'exemple de code suivant ajoute un ARN de SNS rubrique Amazon à my-cluster.

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Pour plus d'informations, consultez UpdateCluster.

Ajouter un SNS sujet Amazon (MemoryDBAPI)

Pour ajouter ou mettre à jour une SNS rubrique Amazon pour un cluster, lancez l'`UpdateCluster`action avec les paramètres suivants :

- `ClusterName=my-cluster`
- `SnsTopicArn=arn%3Aaws%3Asns%3Aus-east-1%3A565419523791%3AmemorydbNotifications`

Pour ajouter ou mettre à jour une SNS rubrique Amazon pour un cluster, lancez l'`UpdateCluster`action.

Pour plus d'informations, consultez UpdateCluster.

Activation et désactivation des notifications Amazon SNS

Vous pouvez activer ou désactiver les notifications pour un cluster. Les procédures suivantes vous montrent comment désactiver les SNS notifications Amazon.

Activation et désactivation SNS des notifications Amazon (console)

Pour désactiver les SNS notifications Amazon à l'aide du AWS Management Console

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse. https://console.aws.amazon.com/memorydb/
2. Cliquez sur le bouton radio situé à gauche du cluster pour lequel vous souhaitez modifier la notification.
3. Sélectionnez Modifier.
4. Dans Modifier le cluster sous Sujet de SNS notification, choisissez Désactiver les notifications.
5. Sélectionnez Modifier.

Activation et désactivation des SNS notifications Amazon ()AWS CLI

Pour désactiver SNS les notifications Amazon, utilisez la commande `update-cluster` avec les paramètres suivants :

Pour Linux, macOS ou Unix :

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --sns-topic-status inactive
```

Pour Windows :

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --sns-topic-status inactive
```

Activation et désactivation des SNS notifications Amazon (APIMemoryDB)

Pour désactiver SNS les notifications Amazon, lancez l'`UpdateCluster`action avec les paramètres suivants :

- `ClusterName=my-cluster`
- `SnsTopicStatus=inactive`

Cet appel vous renvoie des informations semblables à ce qui suit :

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &ClusterName=my-cluster
    &SnsTopicStatus=inactive
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
```

```
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Affichage des événements MemoryDB

MemoryDB enregistre les événements liés à vos clusters, groupes de sécurité et groupes de paramètres. Ces informations comprennent la date et l'heure de l'événement, le nom et le type de la source de l'événement, ainsi qu'une description de cet événement. Vous pouvez facilement récupérer les événements du journal à l'aide de la console MemoryDB, de la AWS CLI `describe-events` commande ou de l'action API MemoryDB. `DescribeEvents`

Les procédures suivantes vous montrent comment afficher tous les événements MemoryDB des dernières 24 heures (1440 minutes).

Affichage des événements MemoryDB (console)

La procédure suivante affiche les événements à l'aide de la console MemoryDB.

Pour afficher les événements à l'aide de la console MemoryDB

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse. https://console.aws.amazon.com/memorydb/
2. Dans le volet de navigation de gauche, sélectionnez Events.

   L'écran Événements affiche la liste de tous les événements disponibles. Chaque ligne de la liste représente un événement et affiche la source de l'événement, le type d'événement (tel que cluster, groupe de paramètres, acl, groupe de sécurité ou groupe de sous-réseaux), l'GMTheure de l'événement et la description de l'événement.

   A l'aide du Filtre, vous pouvez choisir d'afficher tous les événements ou uniquement ceux d'un type spécifique dans la liste des événements.

Affichage des événements MemoryDB ()AWS CLI

Pour générer une liste d'événements MemoryDB à l'aide de AWS CLI, utilisez la commande. `describe-events` Vous pouvez utiliser des paramètres facultatifs pour contrôler le type et la période des événements répertoriés, le nombre maximal d'événements à répertorier, etc.

Le code suivant répertorie jusqu'à 40 événements de cluster.

```
aws memorydb describe-events --source-type cluster --max-results 40
```

Le code suivant répertorie tous les événements qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

```
aws memorydb describe-events --duration 1440
```

La sortie de la commande `describe-events` ressemble à ceci.

```
{
    "Events": [
        {
            "Date": "2021-03-29T22:17:37.781Z",
            "Message": "Added node 0001 in Availability Zone us-east-1a",
            "SourceName": "memorydb01",
            "SourceType": "cluster"
        },
        {
            "Date": "2021-03-29T22:17:37.769Z",
            "Message": "cluster created",
            "SourceName": "memorydb01",
            "SourceType": "cluster"
        }
    ]
}
```

Pour plus d'informations, notamment sur les paramètres disponibles et les valeurs de paramètre autorisées, consultez `describe-events`.

Affichage des événements MemoryDB (MemoryDB) API

Pour générer une liste d'événements MemoryDB à l'aide de MemoryDBAPI, utilisez l'action. `DescribeEvents` Vous pouvez utiliser des paramètres facultatifs pour contrôler le type et la période des événements répertoriés, le nombre maximal d'événements à répertorier, etc.

Le code suivant répertorie les 40 événements -cluster les plus récents.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeEvents
   &MaxResults=40
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cluster
```

```
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Le code suivant répertorie les événements du cluster survenus au cours des dernières 24 heures (1 440 minutes).

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeEvents
   &Duration=1440
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cluster
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Les actions ci-dessus doivent produire un résultat similaire à ce qui suit :

```
<DescribeEventsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
    <DescribeEventsResult>
        <Events>
            <Event>
                <Message>cluster created</Message>
                <SourceType>cluster</SourceType>
                <Date>2021-08-02T18:22:18.202Z</Date>
                <SourceName>my-memorydb-primary</SourceName>
            </Event>

 (...output omitted...)

        </Events>
    </DescribeEventsResult>
    <ResponseMetadata>
        <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
    </ResponseMetadata>
</DescribeEventsResponse>
```

Pour plus d'informations, notamment sur les paramètres disponibles et les valeurs de paramètre autorisées, consultez `DescribeEvents`.

Notifications d'événements Amazon SNS

MemoryDB peut publier des messages à l'aide d'Amazon Simple Notification Service (SNS) lorsque des événements importants se produisent sur un cluster. Cette fonctionnalité peut être utilisée pour actualiser les listes de serveurs sur les machines clientes connectées aux points de terminaison des nœuds individuels d'un cluster.

> **Note**
>
> Pour plus d'informations sur Amazon Simple Notification Service (SNS), y compris des informations sur la tarification et des liens vers la documentation Amazon SNS, veuillez consulter Page produit Amazon SNS.

Les notifications sont publiées sur une rubrique Amazon SNS spécifiée. Ci-après les exigences concernant les notifications :

- Un seul sujet peut être configuré pour les notifications MemoryDB.
- Le AWS compte propriétaire de la rubrique Amazon SNS doit être le même que celui qui possède le cluster sur lequel les notifications sont activées.

Événements MemoryDB

Les événements MemoryDB suivants déclenchent les notifications Amazon SNS :

Nom de l'événement	Message	Description
Base de données de mémoire : AddNodeComplete	`"Modified number of nodes from %d to %d"`	Un nœud a été ajouté au cluster et est prêt à être utilisé.
MemoryDB : AddNodeFailed en raison d'un nombre insuffisant d'adresses IP libres	`"Failed to modify number of nodes from %d to %d due to insufficient free IP addresses"`	Impossible d'ajouter un nœud car il n'y a pas suffisamment d'adresses IP disponibles.

Nom de l'événement	Message	Description
Base de données de mémoire : ClusterParametersChanged	`"Updated parameter group for the cluster"` Dans le cas d'une création, envoyez également `"Updated to use a ParameterGroup %s"`	Un ou plusieurs paramètres de cluster ont été modifiés.
Base de données de mémoire : ClusterProvisioningComplete	`"Cluster created."`	Le provisionnement d'un cluster est terminé et les nœuds du cluster sont prêts à être utilisés.
MemoryDB : ClusterProvisioningFailed en raison d'un état réseau incompatible	`"Failed to create cluster due to incompatible network state. %s"`	Une tentative a été faite pour lancer un nouveau cluster dans un cloud privé virtuel (VPC) inexistant.
Base de données de mémoire : ClusterRestoreFailed	`"Restore from %s failed for node %s. %s"`	MemoryDB n'a pas pu remplir le cluster avec les données de capture instantanée Redis OSS. Cela peut être dû à un fichier instantané inexistant dans Amazon S3 ou à des autorisations incorrectes sur ce fichier. Si vous décrivez le cluster, son statut sera `restore-failed` . Vous devrez supprimer le cluster et recommencer à zéro. Pour plus d'informations, consultez Création d'un nouveau cluster avec un instantané créé en externe.

Nom de l'événement	Message	Description
Base de données de mémoire : ClusterScalingComplete	`"Succeeded applying modification to node type to %s."`	Mise à l'échelle pour que le cluster soit terminé avec succès.
Base de données de mémoire : ClusterScalingFailed	`"Failed applying modification to node type to %s."`	L'opération de mise à l'échelle sur le cluster a échoué.
Base de données de mémoire : ClusterSecurityGroupModified	`"Modified security group for cluster."`	L'un des événements suivants s'est produit : • La liste des groupes de sécurité autorisés pour le cluster a été modifiée. • Un ou plusieurs nouveaux groupes de sécurité EC2 ont été autorisés sur l'un des groupes de sécurité associés au cluster. • Un ou plusieurs groupes de sécurité EC2 ont été révoqués de l'un des groupes de sécurité associés au cluster.

Nom de l'événement	Message	Description
Base de données de mémoire : NodeReplaceStarted	`"Recovering node %s"`	MemoryDB a détecté que l'hôte exécutant un nœud est dégradé ou inaccessible et a commencé à remplacer le nœud. **Note** L'entrée DNS du nœud remplacé n'est pas modifiée. Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Cependant, certaines bibliothèques clientes peuvent cesser d'utiliser le nœud même après que MemoryDB l'ait remplacé ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.

Nom de l'événement	Message	Description
Base de données de mémoire : NodeReplaceComplete	`"Finished recovery for node %s"`	MemoryDB a détecté que l'hôte exécutant un nœud est dégradé ou inaccessible et a terminé de remplacer le nœud. **Note** L'entrée DNS du nœud remplacé n'est pas modifiée. Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Cependant, certaines bibliothèques clientes peuvent cesser d'utiliser le nœud même après que MemoryDB l'ait remplacé ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.
Base de données de mémoire : CreateClusterComplete	`"Cluster created"`	Le cluster a été créé avec succès.

Nom de l'événement	Message	Description
Base de données de mémoire : CreateClusterFailed	`"Failed to create cluster due to unsuccessful creation of its node(s)."` et `"Deleting all nodes belonging to this cluster."`	Le cluster n'a pas été créé.
Base de données de mémoire : DeleteClusterComplete	`"Cluster deleted."`	La suppression d'un cluster et de tous les nœuds associés est terminée.
Base de données de mémoire : FailoverComplete	`"Failover to replica node %s completed"`	Basculement vers un nœud du réplica réussi.
Base de données de mémoire : NodeReplacementCanceled	`"The replacement of node %s which was scheduled during the maintenance window from start time: %s, end time: %s has been canceled"`	Le remplacement d'un nœud de votre cluster qui était prévu a été annulé.
Base de données de mémoire : NodeReplacementRescheduled	`"The replacement in maintenance window for node %s has been re-scheduled from previous start time: %s, previous end time: %s to new start time: %s, new end time: %s"`	Le remplacement d'un nœud de cluster a été reprogrammé dans le créneau indiqué dans la notification. Pour plus d'informations sur les actions que vous pouvez effectuer, consultez Remplacement de nœuds.

Nom de l'événement	Message	Description
Base de données de mémoire : NodeReplacementScheduled	`"The node %s is scheduled for replacement during the maintenance window from start time: %s to end time: %s"`	Un nœud du cluster doit être remplacé pendant le créneau décrit dans la notification. Pour plus d'informations sur les actions que vous pouvez effectuer, consultez Remplacement de nœuds.
Base de données de mémoire : RemoveNodeComplete	`"Removed node %s"`	Un nœud a été supprimé du cluster.
Base de données de mémoire : SnapshotComplete	`"Snapshot %s succeeded for node %s"`	Un instantané s'est terminé avec succès.
Base de données de mémoire : SnapshotFailed	`"Snapshot %s failed for node %s"`	Un instantané a échoué. Consultez les événements du cluster pour une cause plus détaillée. Si vous décrivez l'instantané, voyez DescribeSnapshots, le statut sera`failed`.

Journalisation des appels d'API MemoryDB avec AWS CloudTrail

MemoryDB est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans MemoryDB. CloudTrail capture tous les appels d'API pour MemoryDB sous forme d'événements, y compris les appels depuis la console MemoryDB et les appels de code vers les opérations de l'API MemoryDB. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour MemoryDB. 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

qui a été faite à MemoryDB, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

Informations MemoryDB dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans MemoryDB, 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 MemoryDB, créez une trace. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les 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 :

- Vue d'ensemble de la création d'un journal d'activité
- 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 de MemoryDB sont enregistrées par. CloudTrail Par exemple, les appels au `CreateCluster` `DescribeClusters` et les `UpdateCluster` 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 effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.

- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'élément CloudTrail UserIdentity.

Comprendre les entrées du fichier journal MemoryDB

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'`CreateCluster`action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T17:56:46Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "CreateCluster",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.create-cluster",
    "requestParameters": {
        "clusterName": "memorydb-cluster",
        "nodeType": "db.r6g.large",
        "subnetGroupName": "memorydb-subnet-group",
        "aCLName": "open-access"
    },
    "responseElements": {
        "cluster": {
            "name": "memorydb-cluster",
```

```
            "status": "creating",
            "numberOfShards": 1,
            "availabilityMode": "MultiAZ",
            "clusterEndpoint": {
                "port": 6379
            },
            "nodeType": "db.r6g.large",
            "engineVersion": "6.2",
            "enginePatchVersion": "6.2.6",
            "parameterGroupName": "default.memorydb-redis6",
            "parameterGroupStatus": "in-sync",
            "subnetGroupName": "memorydb-subnet-group",
            "tLSEnabled": true,
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:cluster/memorydb-cluster",
            "snapshotRetentionLimit": 0,
            "maintenanceWindow": "tue:06:30-tue:07:30",
            "snapshotWindow": "09:00-10:00",
            "aCLName": "open-access",
            "dataTiering": "false",
            "autoMinorVersionUpgrade": true
        }
    },
    "requestID": "506fc951-9ae2-42bb-872c-98028dc8ed11",
    "eventID": "2ecf3dc3-c931-4df0-a2b3-be90b596697e",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`DescribeClusters`action. Notez que pour tous les appels MemoryDB Describe et List (`Describe*`et`List*`), la `responseElements` section est supprimée et apparaît sous la forme. `null`


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "userName": "john"
    },
    "eventTime": "2021-07-10T18:39:51Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "DescribeClusters",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.describe-clusters",
    "requestParameters": {
        "maxResults": 50,
        "showShardDetails": true
    },
    "responseElements": null,
    "requestID": "5e831993-52bb-494d-9bba-338a117c2389",
    "eventID": "32a3dc0a-31c8-4218-b889-1a6310b7dd50",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

L'exemple suivant montre une entrée de CloudTrail journal qui enregistre une `UpdateCluster` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T19:23:20Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "UpdateCluster",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.update-cluster",
```

```
    "requestParameters": {
        "clusterName": "memorydb-cluster",
        "snapshotWindow": "04:00-05:00",
        "shardConfiguration": {
            "shardCount": 2
        }
    },
    "responseElements": {
        "cluster": {
            "name": "memorydb-cluster",
            "status": "updating",
            "numberOfShards": 2,
            "availabilityMode": "MultiAZ",
            "clusterEndpoint": {
                "address": "clustercfg.memorydb-cluster.cde8da.memorydb.us-
east-1.amazonaws.com",
                "port": 6379
            },
            "nodeType": "db.r6g.large",
            "engineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "parameterGroupName": "default.memorydb-redis6",
            "parameterGroupStatus": "in-sync",
            "subnetGroupName": "memorydb-subnet-group",
            "tLSEnabled": true,
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:cluster/memorydb-cluster",
            "snapshotRetentionLimit": 0,
            "maintenanceWindow": "tue:06:30-tue:07:30",
            "snapshotWindow": "04:00-05:00",
            "autoMinorVersionUpgrade": true,
            "DataTiering": "false"
        }
    },
    "requestID": "dad021ce-d161-4365-8085-574133afab54",
    "eventID": "e0120f85-ab7e-4ad4-ae78-43ba15dee3d8",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateUser`action. Notez que pour les appels MemoryDB contenant des données sensibles, ces données seront supprimées lors de l' CloudTrail événement correspondant, comme indiqué dans la `requestParameters` section ci-dessous.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T19:56:13Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.create-user",
    "requestParameters": {
        "userName": "memorydb-user",
        "authenticationMode": {
            "type": "password",
            "passwords": [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "accessString": "~* &* -@all +@read"
    },
    "responseElements": {
        "user": {
            "name": "memorydb-user",
            "status": "active",
            "accessString": "off ~* &* -@all +@read",
            "aCLNames": [],
            "minimumEngineVersion": "6.2",
            "authentication": {
                "type": "password",
                "passwordCount": 1
            },
```

```
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:user/memorydb-user"
        }
    },
    "requestID": "ae288b5e-80ab-4ff8-989a-5ee5c67cd193",
    "eventID": "ed096e3e-16f1-4a23-866c-0baa6ec769f6",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


Validation de conformité pour MemoryDB

Des auditeurs tiers évaluent la sécurité et la conformité de MemoryDB dans le cadre de plusieurs programmes de AWS conformité. Cela consiste notamment à :

- Norme de sécurité des données de l'industrie des cartes de paiement (PCI DSS). Pour de plus amples informations, consultez PCI DSS.
- Loi sur la transférabilité et l'imputabilité de l'assurance maladie Accord de partenariat (HIPAA BAA). Pour de plus amples informations, consultez Conformité à la loi HIPAA.
- Contrôles du système et de l'organisation (SOC) 1, 2 et 3. Pour de plus amples informations, consultez SOC.
- Programme fédéral de gestion des risques et des autorisations (FedRAMP) modéré. Pour plus d'informations, consultez FedRAMP.
- ISO/IEC 27001:2013, 27017:2015, 27018:2019et ISO/IEC 9001:2015. Pour plus d'informations, consultez les certifications et services AWS ISO et CSA STAR.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformité.

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation de MemoryDB est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides démarrage rapide de la sécurité et de la conformité. Ces guides de déploiement traitent des considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide de règles dans le Guide du développeur AWS Config : AWS Config évalue dans quelle mesure vos configurations de ressources sont conformes aux pratiques internes, aux directives sectorielles 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.
- AWS Audit Manager : ce AWS service vous aide à auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Sécurité de l'infrastructure dans MemoryDB

En tant que service géré, MemoryDB est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à MemoryDB via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Nous recommandons TLS 1.3 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

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

Confidentialité du trafic inter-réseau

MemoryDB utilise les techniques suivantes pour sécuriser vos données et les protéger contre tout accès non autorisé :

- MemoryDB et Amazon VPC et Amazon VPC explique le type de groupe de sécurité dont vous avez besoin pour votre installation.
- API MemoryDB et points de terminaison VPC d'interface ()AWS PrivateLinkvous permet d'établir une connexion privée entre votre VPC et les points de terminaison de l'API MemoryDB.
- Gestion des identités et des accès dans MemoryDB pour attribuer et limiter les actions des utilisateurs, groupes et rôles.

MemoryDB et Amazon VPC et Amazon VPC

Le service Amazon Amazon Virtual Private Cloud (Amazon VPC) définit un réseau virtuel qui ressemble de près à un centre de données classique. Lorsque vous configurez un Virtual Private Cloud (VPC) avec Amazon VPC, vous pouvez sélectionner sa plage d'adresses IP, créer des sous-réseaux et configurer des tables de routage, des passerelles de réseau et des paramètres de sécurité. Vous pouvez également ajouter un cluster au réseau virtuel et contrôler l'accès au cluster à l'aide de groupes de sécurité Amazon VPC.

Cette section explique comment configurer manuellement un cluster MemoryDB dans un VPC. Ces informations sont destinées aux utilisateurs qui souhaitent mieux comprendre comment MemoryDB et Amazon VPC fonctionnent ensemble.

Rubriques

- Comprendre MemoryDB et les VPC
- Modèles d'accès pour accéder à un cluster MemoryDB dans un Amazon VPC
- Création d'un Virtual Private Cloud (VPC)

Comprendre MemoryDB et les VPC

MemoryDB est totalement intégré à Amazon VPC. Pour les utilisateurs de MemoryDB, cela signifie que :

- MemoryDB lance toujours votre cluster dans un VPC.
- Si vous commencez tout juste à utiliserAWS, un VPC par défaut sera automatiquement créé.
- Si vous disposez d'un VPC par défaut et si vous ne spécifiez pas de sous-réseau lors du lancement d'un cluster, ce dernier est lancé dans votre Amazon VPC par défaut.

Pour plus d'informations, consultez la page Comment identifier vos plateformes prises en charge et déterminer si vous disposez d'un VPC par défaut ?.

Avec Amazon VPC, vous pouvez créer un réseau virtuel dans leAWSCloud qui ressemble de près à un centre de données classique. Vous pouvez configurer votre VPC en sélectionnant sa plage d'adresses IP, en créant des sous-réseaux et en configurant des tables de routage, des passerelles de réseau et des paramètres de sécurité.

MemoryDB gère les mises à niveau logicielles, les correctifs, la détection et la résolution des pannes.

Présentation de MemoryDB dans un VPC

1	Un VPC est une partie isolée duAWSCloud qui se voit attribuer son propre bloc d'adresses IP.
2	Une passerelle Internet connecte votre VPC directement à Internet et fournit un accès à d'autresAWSdes ressources telles qu'Amazon Simple Storage Service (Amazon S3) qui s'exécutent en dehors de votre VPC.
3	Un sous-réseau Amazon VPC est un segment de la plage d'adresses IP d'un VPC où vous pouvez isolerAWSressources en fonction de vos besoins opérationnels et de sécurité.
4	Une table de routage dans votre VPC dirige le trafic réseau entre le sous-réseau et le réseau Internet. L'Amazon VPC possède un routeur implicite.
5	

	Un groupe de sécurité Amazon VPC contrôle le trafic entrant et sortant de vos clusters MemoryDB et les instances Amazon EC2.
6	Vous pouvez lancer un cluster MemoryDB dans le sous-réseau. Les nœuds ont des adresses IP privées provenant de la plage d'adresses du sous-réseau.
7	Vous pouvez également lancer des instances Amazon EC2 dans le sous-réseau. Chaque instance Amazon EC2 dispose d'une adresse IP privée provenant de la plage d'adresse s du sous-réseau. L'instance Amazon EC2 peut se connecter à un nœud dans le même sous-réseau.
8	Pour qu'une instance Amazon EC2 dans votre VPC soit accessible depuis Internet, vous devez affecter une adresse statique, publique appelée « adresse IP élastique » à l'instance concernée.

Prérequis

Pour créer un cluster MemoryDB dans un VPC, votre VPC doit répondre aux exigences suivantes :

- Votre VPC doit autoriser les instances Amazon EC2 non dédiées. Vous ne pouvez pas utiliser MemoryDB dans un VPC qui est configuré pour la location de l'instance dédiée.
- Un groupe de sous-réseaux doit être défini pour votre VPC. MemoryDB utilise ce groupe de sous-réseaux pour sélectionner un sous-réseau et des adresses IP au sein de celui-ci à associer à vos nœuds.
- Un groupe de sécurité doit être défini pour votre VPC ou vous pouvez utiliser la valeur par défaut fournie.
- Les blocs d'adresses CIDR pour chaque sous-réseau doivent être suffisamment grands pour fournir des adresses IP inutilisées pour MemoryDB à utiliser lors des opérations de maintenance.

Routage et sécurité

Vous pouvez configurer le routage dans votre VPC pour contrôler le flux du trafic (par exemple, vers la passerelle Internet ou une passerelle réseau privé virtuel). Avec une passerelle Internet, votre VPC a un accès direct à d'autresAWSressources qui ne s'exécutent pas dans votre VPC. Si vous choisissez de n'avoir qu'une passerelle réseau privé virtuel avec une connexion à votre réseau local,

vous pouvez acheminer votre trafic Internet via le réseau VPN et utiliser les politiques de sécurité et le pare-feu pour contrôler le trafic. Dans ce cas, vous devrez payer de frais supplémentaires liés à la bande passante lorsque vous accédezAWSressources sur Internet.

Vous pouvez utiliser des groupes de sécurité Amazon VPC pour sécuriser les clusters MemoryDB et les instances Amazon EC2 dans votre Amazon VPC. Les groupes de sécurité agissent comme un pare-feu au niveau de l'instance, et non au niveau du sous-réseau.

> **Note**
>
> Nous vous recommandons fortement d'utiliser des noms DNS pour vous connecter à vos nœuds, car l'adresse IP sous-jacente peut changer au fil du temps.

Documentation Amazon VPC

Amazon VPC a son propre ensemble de documentation pour décrire comment créer et utiliser votre Amazon VPC. Le tableau suivant indique où trouver des informations dans les guides Amazon VPC.

Description	Documentation
Commencer à utiliser Amazon VPC	Démarrage avec Amazon VPC
Utilisation d'Amazon VPC via la AWS Management Console	Amazon VPC User Guide
Description complète de toutes les commandes Amazon VPC	Référence des commandes en ligne Amazon EC2 (les commandes Amazon VPC se trouvent dans la référence Amazon EC2)
Descriptions complètes des opérations de l'API Amazon VPC, des types de données et des erreurs	Référence de l’API Amazon EC2 (les opération s d’API Amazon VPC se trouvent dans la référence Amazon EC2)
Informations pour l'administrateur de réseau qui a besoin de configurer la passerelle de votre côté avec une connexion VPN IPsec facultative	Description de AWS Site-to-Site VPN

Pour plus d'informations sur Amazon Virtual Private Cloud, veuillez consulter Amazon Virtual Private Cloud.

Modèles d'accès pour accéder à un cluster MemoryDB dans un Amazon VPC

MemoryDB prend en charge les scénarios suivants pour accéder à un cluster dans un Amazon VPC :

Table des matières

- Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans le même Amazon VPC
- Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents
  - Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents dans la même région
    - Utilisation de Transit Gateway
  - Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans différents VPC Amazon dans différentes régions
    - Utilisation de VPC en transit
- Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client
  - Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN
  - Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans le même Amazon VPC

Le cas d'utilisation le plus courant concerne une application déployée sur une instance EC2 qui doit se connecter à un cluster du même VPC.

La solution la plus simple pour gérer l'accès entre les instances EC2 et les clusters du même VPC consiste à agir ainsi :

1. Créez un groupe de sécurité VPC pour votre cluster. Ce groupe de sécurité peut être utilisé pour restreindre l'accès aux clusters. Par exemple, vous pouvez créer une règle personnalisée pour ce groupe de sécurité, qui autorise l'accès TCP à l'aide du port que vous avez attribué au cluster lorsque vous l'avez créé et une adresse IP que vous utiliserez pour accéder au cluster.

   Le port par défaut pour les clusters MemoryDB est. `6379`

2. Créez un groupe de sécurité VPC pour vos instances EC2 (serveurs web et d'application). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l'instance EC2 à partir d'Internet via la table de routage du VPC. Par exemple, vous pouvez définir des règles sur ce groupe de sécurité pour autoriser l'accès TCP à l'instance EC2 sur le port 22.
3. Créez des règles personnalisées dans le groupe de sécurité de votre cluster qui autorisent les connexions à partir du groupe de sécurité que vous avez créé pour vos instances EC2. N'importe quel membre du groupe de sécurité peut ainsi accéder aux clusters.

Pour créer une règle dans un groupe de sécurité VPC qui autorise les connexions à partir d'un autre groupe de sécurité

1. Connectez-vous à la console de AWS gestion et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc.
2. Dans le volet de navigation de gauche, sélectionnez Security Groups.
3. Sélectionnez ou créez un groupe de sécurité que vous utiliserez pour vos clusters. Sous Règles entrantes, sélectionnez Modifier les règles entrantes, puis Ajouter une règle. Ce groupe de sécurité autorisera l'accès aux membres d'un autre groupe de sécurité.
4. Dans Type, choisissez Règle TCP personnalisée.

    a. Pour Plage de ports, spécifiez le port utilisé lors de la création de votre cluster.

    Le port par défaut pour les clusters MemoryDB est. 6379

    b. Dans le champ Source, saisissez l'ID de votre groupe de sécurité. Dans la liste, sélectionnez le groupe de sécurité que vous utiliserez pour vos instances Amazon EC2.
5. Choisissez Enregistrer lorsque vous avez terminé.

Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents

Lorsque votre cluster se trouve dans un VPC différent de celui de l'instance EC2 que vous utilisez pour y accéder, il existe plusieurs manières d'accéder au cluster. Si le cluster et l'instance EC2 se trouvent dans des VPC différents mais dans la même région, vous pouvez utiliser le peering VPC. Si le cluster et l'instance EC2 se trouvent dans des régions différentes, vous pouvez créer une connectivité VPN entre les régions.

Rubriques

- Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents dans la même région
- Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans différents VPC Amazon dans différentes régions

Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents dans la même région

Cluster auquel accède une instance Amazon EC2 dans un Amazon VPC différent de la même région : connexion d'appairage de VPC

Une connexion d'appairage de VPC est une connexion de mise en réseau entre deux VPC qui permet de router le trafic entre ces derniers à l'aide d'adresses IP privées. Les instances des deux VPC peuvent communiquer entre elles comme si elles se trouvaient dans le même réseau. Vous pouvez créer une connexion d'appairage VPC entre vos propres Amazon VPC ou avec un Amazon VPC d'un autre AWS compte au sein d'une même région. Pour en savoir plus sur l'appairage d'Amazon VPC, veuillez consulter la documentation VPC.

Pour accéder à un cluster dans un Amazon VPC différent via l'appairage

1. Veillez à ce que les plages IP des deux VPC ne se chevauchent pas ou vous ne pourrez pas les appairer.
2. Appairez les deux VPC. Pour de plus amples informations, veuillez consulter Création et acceptation d'une connexion d'appairage d'Amazon VPC.
3. Mettez à jour votre table de routage. Pour de plus amples informations, veuillez consulter Mise à jour de vos tables de routage pour une connexion d'appairage de VPC.
4. Modifiez le groupe de sécurité de votre cluster MemoryDB pour autoriser les connexions entrantes depuis le groupe de sécurité des applications dans le VPC homologue. Pour de plus amples informations, veuillez consulter Référencer des groupes de sécurité du VPC pair.

L'accès à un cluster sur une connexion d'appairage entraînera des frais de transfert de données supplémentaires.

Utilisation de Transit Gateway

Une passerelle de transit vous permet de connecter des VPC et des connexions VPN dans la même AWS région et d'acheminer le trafic entre eux. Une passerelle de transit fonctionne sur plusieurs AWS comptes, et vous pouvez utiliser AWS Resource Access Manager pour partager votre passerelle de transit avec d'autres comptes. Une fois que vous avez partagé une passerelle de transit avec un autre AWS compte, le propriétaire du compte peut associer ses VPC à votre passerelle de transit. Un utilisateur de l'un des comptes peut supprimer l'attachement à tout moment.

Vous pouvez activer la multicast sur une passerelle de transit, puis créer un domaine multicast de passerelle de transit qui autorise l'envoi du trafic multicast à partir de votre source multicast vers des membres de groupe multicast sur des attachements de VPC que vous associez au domaine.

Vous pouvez également créer une pièce jointe de connexion d'appairage entre les passerelles de transport en commun de différentes AWS régions. Cela vous permet d'acheminer le trafic entre les attachements des passerelles de transit dans différentes régions.

Pour plus d'informations, consultez Passerelles de transit.

Accès à un cluster MemoryDB lorsque celui-ci et l'instance Amazon EC2 se trouvent dans différents VPC Amazon dans différentes régions

Utilisation de VPC en transit

Une alternative à l'utilisation de l'appairage de VPC, une autre stratégie courante pour connecter plusieurs VPC situés dans différentes zones géographiques et réseaux distants consiste à créer un VPC en transit faisant office de centre de transit dans le réseau mondial. Un VPC en transit simplifie la gestion du réseau et limite le nombre de connexions requises pour connecter plusieurs VPC et réseaux distants. Cette structure permet de gagner du temps et de l'énergie, ainsi que de réduire les coûts. Elle est en effet implémentée virtuellement et évite donc les dépenses traditionnelles liées à l'implantation physique dans un hub de transit de colocalisation ou au déploiement d'un matériel réseau physique.

Connexion sur différents VPC de différentes régions

Une fois le VPC Transit Amazon établi, une application déployée dans un VPC « relais » d'une région peut se connecter à un cluster MemoryDB d'un VPC « relais » d'une autre région.

Pour accéder à un cluster dans un autre VPC au sein d'une autre région AWS

1. Déployez une solution VPC de transit. Pour plus d'informations, veuillez consulter AWS Transit Gateway.
2. Mettez à jour les tables de routage VPC dans l'application et les VPC pour acheminer le trafic via le VGW (Virtual Private Gateway) et l'appliance VPN. En cas de routage dynamique avec le BGP (Border Gateway Protocol), vos routes peuvent être automatiquement propagées.
3. Modifiez le groupe de sécurité de votre cluster MemoryDB pour autoriser les connexions entrantes depuis la plage d'adresses IP des instances d'application. Notez que vous ne pourrez pas référencer le groupe de sécurité du serveur de l'application dans ce scénario.

Le fait d'accéder à un cluster d'une région à une autre entraînera des latences réseau et des frais de transfert de données entre régions supplémentaires.

Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client

Un autre scénario possible est une architecture hybride dans laquelle les clients ou les applications du centre de données du client peuvent avoir besoin d'accéder à un cluster MemoryDB dans le VPC. Ce scénario est également pris en charge, à condition qu'une connectivité existe entre le VPC d'un client et le centre de données, via un VPN ou via Direct Connect.

Rubriques

- Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN
- Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN

Connexion à MemoryDB depuis votre centre de données via un VPN

Pour accéder à un cluster dans un VPC à partir d'une application sur site via une connexion VPN

1. Etablissez une connectivité VPN en ajoutant une passerelle réseau privé virtuel Hardware vers votre VPC. Pour de plus amples informations, veuillez consulter Ajout d'une passerelle réseau privé virtuel Hardware à votre VPC.
2. Mettez à jour la table de routage VPC du sous-réseau sur lequel votre cluster MemoryDB est déployé afin d'autoriser le trafic provenant de votre serveur d'applications sur site. En cas de routage dynamique avec le BGP (Border Gateway Protocol), vos routes peuvent être automatiquement propagées.
3. Modifiez le groupe de sécurité de votre cluster MemoryDB pour autoriser les connexions entrantes depuis les serveurs d'applications locaux.

Le fait d'accéder à un cluster via une connexion VPN entraînera des latences réseau et des frais de transfert de données supplémentaires.

Accès à un cluster MemoryDB à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Connexion à MemoryDB depuis votre centre de données via Direct Connect

Pour accéder à un cluster MemoryDB à partir d'une application exécutée sur votre réseau à l'aide de Direct Connect

1. Etablissez une connectivité Direct Connect. Pour plus d'informations, voir Getting Started with AWS Direct Connect.
2. Modifiez le groupe de sécurité de votre cluster MemoryDB pour autoriser les connexions entrantes depuis les serveurs d'applications locaux.

Le fait d'accéder à un cluster via une connexion DX peut entraîner des latences réseau et des frais de transfert de données supplémentaires.

Création d'un Virtual Private Cloud (VPC)

Dans cet exemple, vous créez un Virtual Private Cloud (VPC) basé sur le service Amazon VPC avec un sous-réseau privé pour chaque zone de disponibilité.

Création d'un VPC (console)

Pour créer un cluster MemoryDB dans un Amazon Virtual Private Cloud dans un Amazon Virtual Private Cloud

1. Connectez-vous à la Console de gestion AWS et ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le tableau de bord VPC, choisissezCréer un VPC.
3. Sous Resources to create (Ressources à créer), choisissez VPC and more (VPC et autres).
4. UNDERNombre de zones de disponibilité (AZ), choisissez le nombre de zones de disponibilité dans lesquelles vous souhaitez lancer vos sous-réseaux.
5. UNDERNombre de sous-réseaux publics, choisissez le nombre de sous-réseaux publics que vous souhaitez ajouter à votre VPC.
6. UNDERNombre de sous-réseaux privés privés, choisissez le nombre de sous-réseaux privés que vous souhaitez ajouter à votre VPC.

   > **Tip**
   >
   > Notez vos identifiants de sous-réseau, et notez lesquels sont publics et privés. Vous aurez besoin de ces informations lorsque vous lancerez vos clusters et ajouterez une instance Amazon EC2 à votre Amazon VPC.

7. Créez un groupe de sécurité Amazon VPC. Vous allez utiliser ce groupe pour votre cluster et votre instance Amazon EC2.

   a. Dans le panneau de navigation de gauche de laAWS Management Console, choisissezGroupes de sécurité.

   b. Sélectionnez Créer un groupe de sécurité.

   c. Entrez un nom et une description pour votre groupe de sécurité dans les zones correspondantes. PourVPC, choisissez l'identifiant de votre VPC.

   d. Lorsque les paramètres vous conviennent, choisissez Yes, Create.

8. Définissez une règle de trafic entrant réseau pour votre groupe de sécurité. Cette règle permet de vous connecter à votre instance Amazon EC2 à l'aide du protocole SSH (Secure Shell).

   a. Dans le volet de navigation de gauche, sélectionnez Security Groups.

   b. Recherchez votre groupe de sécurité dans la liste, puis sélectionnez-le.

   c. Sous Security Group, choisissez l'onglet Inbound. Dans la zone Create a new rule, choisissez SSH, puis Add Rule.

      Définissez les valeurs suivantes pour que votre nouvelle règle entrante autorise HTTP à accéder à :

      - Type : HTTP
      - Source : 0.0.0.0/0

   d. Définissez les valeurs suivantes pour que votre nouvelle règle entrante autorise HTTP à accéder à :

      - Type : HTTP
      - Source : 0.0.0.0/0

      Choisissez Apply Rule Changes.

Maintenant, vous êtes prêt à créer unGroupe de sous-réseauxetcréer un clusterdans votre VPC.

Sous-réseaux et groupes de sous-réseaux

Un groupe de sous-réseaux est un ensemble de sous-réseaux (généralement privés) que vous pouvez utiliser pour vos clusters fonctionnant dans un environnement Amazon Virtual Private Cloud (VPC).

Lorsque vous créez un cluster dans un Amazon VPC, vous pouvez spécifier un groupe de sous-réseaux ou utiliser celui fourni par défaut. MemoryDB utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds.

Cette section explique comment créer et exploiter des sous-réseaux et des groupes de sous-réseaux pour gérer l'accès à vos ressources MemoryDB.

Pour plus d'informations sur l'utilisation du groupe de sous-réseaux dans un environnement Amazon VPC, veuillez consulter Étape 2 : Autoriser l'accès au cluster.

Identifiants MemoryDB AZ pris en charge

Nom de région/Région	Identifiants de zone de disponibilité pris en charge		
Région US East (Ohio) `us-east-2`	`use2-az1, use2-az2, use2-az3`		
Région US East (N. Virginia) `us-east-1`	`use1-az2, use1-az4, use1-az6`		
Région US West (N. California) `us-west-1`	`usw1-az1, usw1-az2, usw1-az3`		
Région US West (Oregon) `us-west-2`	`usw2-az1, usw2-az2, usw2-az3`		

Nom de région/Région	Identifiants de zone de disponibilité pris en charge		
Région Canada (Centre) `ca-central-1`	`cac1-az1, cac1-az2, cac1-az4`		
Région Asie-Pacifique (Hong Kong) `ap-east-1`	`ape1-az1, ape1-az2, ape1-az3`		
Région Asia Pacific (Mumbai) `ap-south-1`	`aps1-az1, aps1-az2, aps1-az3`		
Région Asia Pacific (Tokyo) `ap-northeast-1`	`apne1-az1, apne1-az2, apne1-az4`		
Asia Pacific (Seoul) Region `ap-northeast-2`	`apne2-az1, apne2-az2, apne2-az3`		
Région Asie-Pacifique (Singapour) `ap-southeast-1`	`apse1-az1, apse1-az2, apse1-az3`		
Région Asia Pacific (Sydney) `ap-southeast-2`	`apse2-az1, apse2-az2, apse2-az3`		

Nom de région/Région	Identifiants de zone de disponibilité pris en charge		
Région Europe (Frankfurt) `eu-central-1`	`euc1-az1, euc1-az2, euc1-az3`		
Région Europe (Irlande) `eu-west-1`	`euw1-az1, euw1-az2, euw1-az3`		
Région Europe (London) `eu-west-2`	`euw2-az1, euw2-az2, euw2-az3`		
Région Europe (Paris) `eu-west-3`	`euw3-az1, euw3-az2, euw3-az3`		
Région Europe (Stockholm) `eu-north-1`	`eun1-az1, eun1-az2, eun1-az3`		
Europe (Milan) Region `eu-south-1`	`eus1-az1, eus1-az2, eus1-az3`		
Région South America (São Paulo) `sa-east-1`	`sae1-az1, sae1-az2, sae1-az3`		

Nom de région/Région	Identifiants de zone de disponibilité pris en charge		
Région Chine (Beijing) `cn-north-1`	`cnn1-az1, cnn1-az2`		
Région Chine (Ningxia) `cn-northwest-1`	`cnw1-az1, cnw1-az2, cnw1-az3`		

Rubriques

- Création d'un groupe de sous-réseaux
- Mettre à jour un groupe de sous-réseaux
- Afficher les détails d'un groupe de sous-réseaux
- Suppression d'un groupe de sous-réseaux

Création d'un groupe de sous-réseaux

Lorsque vous créez un nouveau groupe de sous-réseaux de , notez le nombre d'adresses IP disponibles. Si le sous-réseau a très peu d'adresses IP libres, vous pourriez ne pas pouvoir ajouter autant de nœuds de que vous le souhaitez au cluster. Pour résoudre ce problème, vous pouvez assigner un ou plusieurs sous-réseaux à un groupe de sous-réseaux afin d'avoir un nombre suffisant d'adresses IP dans la zone de disponibilité de votre cluster. Vous pouvez, ensuite, ajouter plusieurs nœuds de cache à votre cluster.

Les procédures suivantes vous montrent comment créer un groupe de sous-réseaux appelé `mysubnetgroup` (console), le AWS CLI, et l'API MemoryDB.

Pour créer un groupe de sous-réseaux (console)

La procédure suivante indique comment créer un groupe de sous-réseaux (console).

Pour créer un groupe de sous-réseaux (console)

1. Connectez-vous à la console de AWS gestion et ouvrez la console MemoryDB à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Subnet Groups.
3. Choisissez Créer un groupe de sous-réseaux.
4. Sur la page Créer un groupe de sous-réseaux, procédez comme suit :

    a. Dans le champ Name, saisissez le nom de votre groupe de sous-réseaux de

        Les contraintes d'attribution de noms de cluster sont les suivantes :

        - Doit contenir entre 1 et 40 caractères alphanumériques ou traits d'union.
        - Doit commencer par une lettre.
        - Ils ne peuvent pas comporter deux traits d'union consécutifs.
        - Ils ne peuvent pas se terminer par un trait d'union.

    b. Dans la zone Description, saisissez une description de votre groupe de sous-réseaux de

    c. Dans la zone VPC ID (ID du VPC), choisissez l'Amazon VPC que vous avez créé. Si vous n'en avez pas créé un, cliquez sur le bouton Créer un VPC et suivez les étapes pour en créer un.

    d. Dans Sous-réseaux sélectionnés, choisissez la zone de disponibilité et l'ID de votre sous-réseau privé, puis choisissez Choisir.

5. Pour les balises, vous pouvez éventuellement appliquer des balises pour rechercher et filtrer vos sous-réseaux ou suivre vos AWS coûts.
6. Lorsque tous les paramètres vous conviennent, choisissez Créer.
7. Dans le message de confirmation qui s'affiche, cliquez sur Close.

Votre nouveau groupe de sous-réseaux apparaît dans la liste des groupes de sous-réseaux de la console MemoryDB. En bas de la fenêtre, vous pouvez choisir le groupe de sous-réseaux pour voir les détails, tels que tous les sous-réseaux associés à ce groupe.

Création d'un groupe de sous-réseaux (AWS CLI)

A l'invite de commande, utilisez la commande `create-subnet-group` pour créer un groupe de sous-réseaux de

Pour Linux, macOS ou Unix :

```
aws memorydb create-subnet-group \
    --subnet-group-name mysubnetgroup \
    --description "Testing" \
    --subnet-ids subnet-53df9c3a
```

Pour Windows :

```
aws memorydb create-subnet-group ^
    --subnet-group-name mysubnetgroup ^
    --description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
    {
        "SubnetGroup": {
            "Subnets": [
                {
                    "Identifier": "subnet-53df9c3a",
                    "AvailabilityZone": {
                    "Name": "us-east-1a"
                    }
                }
            ],
```

```
            "VpcId": "vpc-3cfaef47",
            "Name": "mysubnetgroup",
            "ARN": "arn:aws:memorydb:us-east-1:012345678912:subnetgroup/
mysubnetgroup",
            "Description": "Testing"
        }
    }
```

Pour plus d'informations, consultez la AWS CLI rubriquecreate-subnet-group.

Création d'un groupe de sous-réseaux (API MemoryDB)

À l'aide de l'API MemoryDB, appelez `CreateSubnetGroup` avec les paramètres suivants :

- `SubnetGroupName=`*`mysubnetgroup`*
- `Description=`*`Testing`*
- `SubnetIds.member.1=`*`subnet-53df9c3a`*

Mettre à jour un groupe de sous-réseaux

Vous pouvez mettre à jour la description d'un groupe de sous-réseaux ou modifier la liste des ID de sous-réseaux associés au groupe de sous-réseaux. Vous ne pouvez pas supprimer un ID de sous-réseau d'un groupe de sous-réseaux de si un cluster de utilise actuellement ce sous-réseau.

Les procédures suivantes indiquent comment mettre à jour un groupe de sous-réseaux.

Mise à jour de groupes de sous-réseaux (console)

Pour mettre à jour un groupe de sous-réseaux

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Subnet Groups.
3. Dans la liste des groupes de sous-réseaux, choisissez celui que vous voulez modifier.
4. Les champs Nom, VpcID et Description ne sont pas modifiables.
5. Dans la section Sous-réseaux sélectionnés, cliquez sur Gérer pour apporter les modifications nécessaires aux zones de disponibilité dont vous avez besoin pour les sous-réseaux. Choisissez Save pour enregistrer les changements.

Mise à jour de groupes de sous-réseaux (AWS CLI)

À l'invite de commande, utilisez la commande `update-subnet-group` pour mettre à jour un groupe de sous-réseaux.

Pour Linux, macOS ou Unix :

```
aws memorydb update-subnet-group \
    --subnet-group-name mysubnetgroup \
    --description "New description" \
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Pour Windows :

```
aws memorydb update-subnet-group ^
    --subnet-group-name mysubnetgroup ^
    --description "New description" ^
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "SubnetGroup": {
        "VpcId": "vpc-73cd3c17",
        "Description": "New description",
        "Subnets": [
            {
                "Identifier": "subnet-42dcf93a",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            },
            {
                "Identifier": "subnet-48fc12a9",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "Name": "mysubnetgroup",
        "ARN": "arn:aws:memorydb:us-east-1:012345678912:subnetgroup/mysubnetgroup",
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubrique update-subnet-group.

Mise à jour de groupes de sous-réseaux (API MemoryDB)

À l'aide de l'API MemoryDB, appelez `UpdateSubnetGroup` avec les paramètres suivants :

- `SubnetGroupName=`*`mysubnetgroup`*
- D'autres paramètres dont vous voulez modifier les valeurs. Cet exemple utilise `Description=`*`New%20description`* pour modifier la description du groupe de sous-réseaux de

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateSubnetGroup
    &Description=New%20description
    &SubnetGroupName=mysubnetgroup
    &SubnetIds.member.1=subnet-42df9c3a
```

```
    &SubnetIds.member.2=subnet-48fc21a9
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20141201T220302Z
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

> **Note**
>
> Lorsque vous créez un nouveau groupe de sous-réseaux de , notez le nombre d'adresses IP disponibles. Si le sous-réseau a très peu d'adresses IP libres, vous pourriez ne pas pouvoir ajouter autant de nœuds de que vous le souhaitez au cluster. Pour résoudre ce problème, vous pouvez assigner un ou plusieurs sous-réseaux à un groupe de sous-réseaux afin d'avoir un nombre suffisant d'adresses IP dans la zone de disponibilité de votre cluster. Vous pouvez, ensuite, ajouter plusieurs nœuds de cache à votre cluster.

Afficher les détails d'un groupe de sous-réseaux

Les procédures suivantes vous montrent comment afficher les détails d'un groupe de sous-réseaux.

Affichage des détails des groupes de sous-réseaux (console)

Pour afficher les détails d'un groupe de sous-réseaux (console)

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Subnet Groups.
3. Sur la page Groupes de sous-réseaux, choisissez le groupe de sous-réseaux sous Nom ou entrez le nom du groupe de sous-réseaux dans la barre de recherche.
4. Sur la page Groupes de sous-réseaux, choisissez le groupe de sous-réseaux sous Nom ou entrez le nom du groupe de sous-réseaux dans la barre de recherche.
5. Dans les paramètres du groupe de sous-réseaux, vous pouvez afficher le nom, la description, l'ID VPC et le nom de ressource Amazon (ARN) du groupe de sous-réseaux.

6. Sous Sous-réseaux, vous pouvez afficher les zones de disponibilité, les ID de sous-réseau et les blocs CIDR du groupe de sous-réseaux.
7. Sous Balises, vous pouvez afficher toutes les balises associées au groupe de sous-réseaux.

Affichage des détails des groupes de sous-réseaux (AWS CLI)

À l'invite de commande, utilisez la commande `describe-subnet-groups` pour afficher les détails d'un groupe de sous-réseaux spécifié.

Pour Linux, macOS ou Unix :

```
aws memorydb describe-subnet-groups \
    --subnet-group-name mysubnetgroup
```

Pour Windows :

```
aws memorydb describe-subnet-groups ^
    --subnet-group-name mysubnetgroup
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
  "subnetgroups": [
    {
      "Subnets": [
        {
          "Identifier": "subnet-060cae3464095de6e",
          "AvailabilityZone": {
            "Name": "us-east-1a"
          }
        },
        {
          "Identifier": "subnet-049d11d4aa78700c3",
          "AvailabilityZone": {
            "Name": "us-east-1c"
          }
        },
        {
          "Identifier": "subnet-0389d4c4157c1edb4",
          "AvailabilityZone": {
            "Name": "us-east-1d"
```

```
          }
        }
      ],
      "VpcId": "vpc-036a8150d4300bcf2",
      "Name": "mysubnetgroup",
      "ARN": "arn:aws:memorydb:us-east-1:53791xzzz7620:subnetgroup/mysubnetgroup",
      "Description": "test"
    }
  ]
}
```

Pour afficher les détails de tous les groupes de sous-réseaux, utilisez la même commande, mais sans spécifier de nom de groupe de sous-réseaux.

```
aws memorydb describe-subnet-groups
```

Pour plus d'informations, consultez la AWS CLI rubriquedescribe-subnet-groups.

Affichage des groupes de sous-réseaux (API MemoryDB)

À l'aide de l'API MemoryDB, appelez `DescribeSubnetGroups` avec les paramètres suivants :

`SubnetGroupName=`*`mysubnetgroup`*

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateSubnetGroup
    &Description=New%20description
    &SubnetGroupName=mysubnetgroup
    &SubnetIds.member.1=subnet-42df9c3a
    &SubnetIds.member.2=subnet-48fc21a9
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20211801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20210801T220302Z
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Signature=<signature>
```

```
    &X-Amz-SignedHeaders=Host
```

Suppression d'un groupe de sous-réseaux

Si vous décidez que vous n'avez plus besoin de votre groupe de sous-réseaux de , vous pouvez le supprimer. Vous ne pouvez pas supprimer un groupe de sous-réseaux de s'il est actuellement utilisé par un cluster de Vous ne pouvez pas non plus supprimer un groupe de sous-réseaux sur un cluster avec Multi-AZ activé si cela laisse ce cluster avec moins de deux sous-réseaux. Vous devez d'abord décocher Multi-AZ, puis supprimer le sous-réseau.

Les procédures suivantes vous montrent comment supprimer un groupe de sous-réseaux.

Suppression d’un groupe de sous-réseaux (console)

Pour supprimer un groupe de sous-réseaux

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le volet de navigation de gauche, choisissez Subnet Groups.
3. Dans la liste des groupes de sous-réseaux, choisissez celui que vous souhaitez supprimer, sélectionnez Actions, puis sélectionnez Supprimer.

   > **Note**
   >
   > Vous ne pouvez pas supprimer un groupe de sous-réseaux par défaut ou un groupe associé à un cluster.

4. L'écran de confirmation de suppression des groupes de sous-réseaux s'affiche.
5. Pour supprimer le groupe de sous-réseaux, entrez `delete` dans la zone de texte de confirmation. Pour conserver le groupe de sous-réseaux, choisissez Cancel (Annuler).

Supprimer un groupe de sous-réseaux (AWS CLI)

À l'aide de AWS CLI, appelez la commande delete-subnet-group avec le paramètre suivant :

- `--subnet-group-name` *`mysubnetgroup`*

Pour Linux, macOS ou Unix :

```
aws memorydb delete-subnet-group \
```

```
    --subnet-group-name mysubnetgroup
```

Pour Windows :

```
aws memorydb delete-subnet-group ^
    --subnet-group-name mysubnetgroup
```

Pour plus d'informations, consultez la AWS CLI rubrique delete-subnet-group.

Supprimer un groupe de sous-réseaux (API MemoryDB)

À l'aide de l'API MemoryDB, appelez `DeleteSubnetGroup` avec le paramètre suivant :

- `SubnetGroupName=mysubnetgroup`

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteSubnetGroup
    &SubnetGroupName=mysubnetgroup
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20210801T220302Z
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la rubrique relative à l'API MemoryDB. DeleteSubnetGroup

API MemoryDB et points de terminaison VPC d'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et les points de terminaison d'API Amazon MemoryDB en créant un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par AWS PrivateLink. AWS PrivateLink vous permet d'accéder en privé

aux opérations de l'API MemoryDB sans passerelle Internet, périphérique NAT, connexion VPN ou connexion Direct AWS Connect.

Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les points de terminaison de l'API MemoryDB. Vos instances n'ont pas non plus besoin d'adresses IP publiques pour utiliser les opérations d'API MemoryDB disponibles. Le trafic entre votre VPC et MemoryDB ne quitte pas le réseau Amazon. Chaque point de terminaison d'interface est représenté par une ou plusieurs interfaces réseau Elastic dans vos sous-réseaux. Pour plus d'informations sur les interfaces réseau Elastic, veuillez consulter Interfaces réseau Elastic dans le Guide de l'utilisateur Amazon EC2.

- Pour plus d'informations sur les points de terminaison VPC, consultez la section Interface VPC endpoints () dans AWS PrivateLink le guide de l'utilisateur Amazon VPC.
- Pour plus d'informations sur les opérations de l'API MemoryDB, consultez la section Opérations de l'API MemoryDB.

Après avoir créé un point de terminaison VPC d'interface, si vous activez les noms d'hôte DNS privés pour le point de terminaison, il s'agit du point de terminaison MemoryDB par défaut (https://memorydb). *Region* .amazonaws.com) correspond à votre point de terminaison VPC. 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.memorydb.Region.vpce.amazonaws.com
```

Pour plus d'informations, consultez Interface VPC Endpoints (AWS PrivateLink) dans le guide de l'utilisateur Amazon VPC. MemoryDB prend en charge les appels à toutes ses actions d'API au sein de votre VPC.

> **Note**
>
> Les noms d'hôtes DNS privés ne peuvent être activés que pour un seul point de terminaison d'un VPC dans le VPC. Si vous voulez créer un point de terminaison d'un VPC supplémentaire, le nom d'hôte DNS privé doit être désactivé pour celui-ci.

Considérations relatives aux points de terminaison d'un VPC

Avant de configurer un point de terminaison VPC d'interface pour les points de terminaison d'API MemoryDB, assurez-vous de consulter les propriétés et les limites du point de terminaison d'interface dans le guide de l'utilisateur Amazon VPC. Toutes les opérations de l'API MemoryDB pertinentes pour la gestion des ressources MemoryDB sont disponibles depuis votre VPC à l'aide de. AWS PrivateLink Les politiques de point de terminaison VPC sont prises en charge pour les points de terminaison de l'API MemoryDB. Par défaut, l'accès complet aux opérations de l'API MemoryDB est autorisé via le point de terminaison. 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.

Création d'un point de terminaison VPC d'interface pour l'API MemoryDB

Vous pouvez créer un point de terminaison VPC pour l'API MemoryDB à l'aide de la console Amazon VPC ou du. AWS CLI Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Une fois que vous avez créé un point de terminaison de VPC d'interface, vous pouvez activer les noms d'hôte DNS privés pour le point de terminaison. Lorsque vous le faites, le point de terminaison MemoryDB par défaut (https://memorydb. *Region* .amazonaws.com) correspond à votre point de terminaison VPC. Pour plus d'informations, consultez Accès à un service via un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Création d'une politique de point de terminaison VPC pour l'API Amazon MemoryDB

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à l'API MemoryDB. La stratégie spécifie les éléments suivants :

- Le principal qui peut exécuter 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 pour les actions de l'API MemoryDB

Voici un exemple de politique de point de terminaison pour l'API MemoryDB. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux actions de l'API MemoryDB répertoriées pour tous les principaux sur toutes les ressources.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "memorydb:CreateCluster",
   "memorydb:UpdateCluster",
   "memorydb:CreateSnapshot"
  ],
  "Resource": "*"
 }]
}
```

Example Politique de point de terminaison VPC qui refuse 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. La politique autorise toutes les actions provenant d'autres comptes.

```
{
 "Statement": [{
   "Action": "*",
   "Effect": "Allow",
   "Resource": "*",
   "Principal": "*"
  },
  {
   "Action": "*",
   "Effect": "Deny",
   "Resource": "*",
   "Principal": {
    "AWS": [
     "123456789012"
    ]
   }
  }
 ]
}
```

Mises à jour du service dans MemoryDB

MemoryDB surveille automatiquement votre parc de clusters et de nœuds pour appliquer les mises à jour de service dès qu'elles sont disponibles. Généralement, vous configurez une fenêtre de maintenance prédéfinie afin que MemoryDB puisse appliquer ces mises à jour. Cependant, dans certains cas, cette approche peut vous sembler trop rigide et être susceptible de restreindre vos flux d'activité.

Avec les mises à jour de service, vous contrôlez quelles mises à jour sont appliquées et à quel moment. Vous pouvez également suivre en temps réel la progression de ces mises à jour sur le cluster MemoryDB sélectionné.

Gestion des mises à jour du service

Les mises à jour du service MemoryDB sont publiées régulièrement. Si vous disposez d'un ou de plusieurs clusters éligibles pour ces mises à jour de service, vous recevez des notifications par e-mail, via les réseaux sociaux, le Personal Health Dashboard (PHD) et les CloudWatch événements Amazon lorsque les mises à jour sont publiées. Les mises à jour sont également affichées sur la page Service Updates de la console MemoryDB. En utilisant ce tableau de bord, vous pouvez consulter toutes les mises à jour du service et leur statut pour votre parc MemoryDB.

Vous contrôlez le moment où vous devez appliquer une mise à jour avant qu'une mise à jour automatique ne démarre. Nous vous recommandons vivement d'appliquer toute mise à jour de type security-update dès que possible afin de garantir que votre MemoryDB contient toujours up-to-date les correctifs de sécurité les plus récents.

Les sections suivantes décrivent ces options en détail :

Rubriques

- Application des mises à jour de service

Application des mises à jour de service

Vous pouvez commencer à appliquer les mises à jour de service à votre flotte à partir du moment où les mises à jour ont un statut available (disponible). Les mises à jour de service sont cumulatives. En d'autres termes, toutes les mises à jour que vous n'avez pas encore appliquées sont incluses dans votre dernière mise à jour.

Si la mise à jour automatique est activée pour une mise à jour de service, vous pouvez choisir de ne prendre aucune mesure lorsqu'elle devient disponible. MemoryDB planifiera d'appliquer la mise à jour pendant la fenêtre de maintenance de vos clusters après la date de début de la mise à jour automatique. Vous recevrez des notifications associées pour chaque étape de la mise à jour.

> **Note**
>
> Vous pouvez appliquer uniquement les mises à jour de service qui ont un statut available (disponible) ou scheduled (planifié).

Pour plus d'informations sur la révision et l'application de mises à jour spécifiques au service aux clusters MemoryDB applicables, consultez. Application des mises à jour du service à l'aide de la console

Lorsqu'une nouvelle mise à jour de service est disponible pour un ou plusieurs de vos clusters MemoryDB, vous pouvez utiliser la console MemoryDB, l'API ou AWS CLI pour appliquer la mise à jour. Les sections suivantes décrivent les options dont vous disposez pour appliquer les mises à jour.

Application des mises à jour du service à l'aide de la console

Pour afficher la liste des mises à jour de service disponibles, ainsi que d'autres informations, accédez à la page Service Updates (Mises à jour de service) dans la console.

1. Connectez-vous à la console MemoryDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/memorydb/.
2. Dans le panneau de navigation, choisissez Service Updates (Mises à jour des services).

Sous Détails de la mise à jour du service, vous pouvez consulter les informations suivantes :

- Nom de la mise à jour de service : le nom unique de la mise à jour de service
- Description de la mise à jour : informations détaillées sur la mise à jour du service
- Date de début de mise à jour automatique : si cet attribut est défini, MemoryDB commencera à planifier la mise à jour automatique de vos clusters dans les fenêtres de maintenance appropriées après cette date. Vous recevrez des notifications à l'avance sur la période de maintenance planifiée exacte, qui peut ne pas être celle qui suit immédiatement la date de début de la mise à jour automatique. Vous pouvez toujours appliquer la mise à jour à vos clusters à tout moment. Si

l'attribut n'est pas défini, la mise à jour du service n'est pas activée et MemoryDB ne mettra pas automatiquement à jour vos clusters.

Dans la section Cluster update status (Statut de mise à jour des clusters), vous pouvez afficher une liste des clusters où la mise à jour de service n'a pas été appliquée ou vient de l'être récemment. Pour chaque cluster, vous pouvez afficher les éléments suivants :

- Nom du cluster : le nom du cluster
- Nœuds mis à jour : ratio des nœuds individuels d'un cluster spécifique qui ont été mis à jour ou qui restent disponibles pour la mise à jour d'un service spécifique.
- Type de mise à jour : le type de la mise à jour de service, qui est soit security-update (mise à jour de sécurité), soit engine-update (mise à jour du moteur)
- Statut : le statut de la mise à jour de service sur le cluster, qui est l'un des suivants :
  - disponible : la mise à jour est disponible pour le cluster requis.
  - en cours d'application : la mise à jour est en cours d'application à ce cluster.
  - scheduled (planifiée) : la date de mise à jour a été programmée.
  - complete (achevée) : la mise à jour a été correctement effectuée. Le cluster dont le statut est complet sera affiché pendant 7 jours après son achèvement.

  Si vous choisissez l'un ou l'ensemble des clusters ayant le statut available (disponible) ou scheduled (planifié), puis choisissez Apply now (Appliquer maintenant), la mise à jour commencera à être appliquée à ces clusters.

Application des mises à jour du service à l'aide de la AWS CLI

Après avoir reçu la notification indiquant que les mises à jour du service sont disponibles, vous pouvez les inspecter et les appliquer à l'aide de la AWS CLI :

- Pour afficher une description des mises à jour de service disponibles, exécutez la commande suivante :

  ```
  aws memorydb describe-service-updates --status available
  ```

  Pour plus d'informations, consultez describe-service-updates.
- Pour appliquer une mise à jour de service à une liste de clusters, exécutez la commande suivante :

```
aws memorydb batch-update-cluster --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

Pour plus d'informations, consultez batch-update-cluster.

Référence

Les rubriques de cette section présente l'utilisation de l'API MemoryDB et de la section MemoryDB de l'.AWS CLI. Sont également présentés les messages d'erreur et les notifications de service courants.

- Utilisation de l'API MemoryDB
- Référence d'API MemoryDB
- Section MemoryDB duAWS CLIRéférence

Utilisation de l'API MemoryDB

Cette section fournit des descriptions de tâches et décrit comment utiliser et mettre en œuvre les opérations MemoryDB. Pour une description complète de ces opérations, consultezRéférence d'API MemoryDB.

Rubriques

- Utilisation de l'API Query
- Bibliothèques disponibles
- Applications de dépannage

Utilisation de l'API Query

Paramètres Query (Requête)

Ces demandes basées sur Query HTTP sont des demandes HTTP qui utilisent le verbe HTTP GET ou POST et un paramètre Query appelé `Action`.

Chaque demande Query doit inclure certains paramètres communs pour gérer l'authentification et la sélection d'une action.

Certaines actions demandent des listes de paramètres. Ces listes sont spécifiées en utilisant la notation `param.`*n*. Les valeurs de *n* sont des nombres entiers à partir de 1.

Authentification de demande Query

Vous pouvez envoyer uniquement des demandes Query via HTTPS, et vous devez inclure une signature dans chaque demande Query. Cette section explique comment créer la signature. La méthode décrite dans la procédure suivante est appelée signature version 4.

Voici les étapes de base utilisées pour authentifier les demandes à AWS. Ce processus suppose que vous êtes enregistré avec AWS et disposez d'un ID de clé d'accès et d'une clé d'accès secrète.

Processus d'authentification des requêtes

1. L'expéditeur crée une demande à AWS.
2. L'expéditeur calcule la signature de la demande, un hachage avec clé pour un code HMAC (code d'authentification d'une empreinte cryptographique de message avec clé) utilisant une fonction de hachage SHA-1, comme défini dans la prochaine section de cette rubrique.

3. L'expéditeur de la demande envoie les données de la demande, la signature et l'ID de clé d'accès (l'identifiant de la clé d'accès secrète utilisée) à AWS.
4. AWS utilise l'identifiant de la clé d'accès pour rechercher la clé d'accès secrète.
5. En appliquant le même algorithme utilisé pour calculer la signature dans la demande, AWS génère une signature à partir des données de la demande et de la clé d'accès secrète.
6. Si la signature correspond, la demande est considérée comme authentique. Si la comparaison échoue, la demande est rejetée, et AWS renvoie une réponse d'erreur.

> **Note**
>
> Si une demande contient un paramètre `Timestamp`, la signature calculée pour la demande expire 15 minutes après sa valeur.
> Si une demande contient un paramètre `Expires`, la signature expire au moment spécifié par le paramètre `Expires`.

Pour calculer la signature de la demande

1. Créez la chaîne de requête de base que vous utiliserez à une étape ultérieure de la procédure :
   a. Triez les composants de la chaîne de requête UTF-8 par nom de paramètre disposé selon l'ordre naturel des octets. Les paramètres peuvent provenir de l'URI GET ou du corps POST (lorsque le type de contenu est application/x-www-form-urlencoded).
   b. URL-encodez le nom et les valeurs du paramètre en appliquant les règles suivantes :
      i. Ne pas URL-encoder les caractères que le RFC définit comme autorisés. Les caractères autorisés sont A à Z, a à z, 0 à 9, le trait d'union (-), le trait de soulignement (_), le point final (.) et le tilde (~).
      ii. %-encodez tous les autres caractères avec %XY, où X et Y représentent les caractères hexadécimaux 0 à 9 et les lettres majuscules A à F.
      iii. %-encodez les caractères UTF-8 étendus dans la forme %XY%ZA....
      iv. %-encodez le caractère espace en %20 (et non pas en +, comme le font les schémas d'encodage courants).
   c. Utilisez le symbole équivalent (=) (ASCII caractère 61) pour séparer les noms de paramètres codés de leurs valeurs codées, même si la valeur du paramètre est vide.

d. Séparez les paires nom-valeur en insérant une esperluette (&) (code ASCII 38).

2. Créez la chaîne de connexion en appliquant la grammaire suivante (le « \n » représente une nouvelle ligne ASCII).

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

Le composant HTTPRequestURI est le composant du chemin absolu HTTP de l'URI menant jusqu'à la chaîne de requête de demandes sans cependant l'inclure. Si le composant HTTPRequestURI est vide, utilisez une barre oblique (/).

3. Définissez un HMAC conforme à RFC 2104 à l’aide de la chaîne que vous venez de créer, votre clé d’accès secrète comme clé et l’algorithme haché SHA256 ou SHA1.

Pour plus d'informations, consultez https://www.ietf.org/rfc/rfc2104.txt.

4. Convertissez la valeur qui est générée en Base64.

5. Incluez la valeur comme la valeur du paramètre `Signature` dans la demande.

Par exemple, voici un exemple de demande (sauts de ligne ajoutés pour plus de clarté).

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2021-01-01
```

Pour la chaîne de requête précédente, vous devez calculer la signature HMAC sur la chaîne suivante.

```
GET\n
    memory-db.amazonaws.com\n
    Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
```

```
    &SignatureVersion=4
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-east-1%2Fmemorydb%2Faws4_request
    &X-Amz-Date=20210801T223649Z
    &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-
amz-date
        content-type:
        host:memory-db.us-east-1.amazonaws.com
        user-agent:ServicesAPICommand_Client
    x-amz-content-sha256:
    x-amz-date:
```

Le résultat est la demande signée suivante.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-east-1/memorydb/aws4_request
    &X-Amz-Date=20210801T223649Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

Pour plus d'informations sur le processus de signature et le calcul de la signature de la demande, consultez la rubriqueProcessus de signature Signature Version 4et ses sous-sujets.

Bibliothèques disponibles

AWS fournit les kits de développement logiciel (SDK) pour les développeurs qui veulent créer des applications à l'aide d'API langage spécifique au lieu de l'API Query. Ces kits de développement logiciel (SDK) offrent des fonctions de base (non présentes dans les API), telles que l'authentification de demande, les nouvelles tentatives de demande et la gestion des erreurs ; celles-ci vous permettent de démarrer plus facilement. Des kits de développement logiciel et des ressources supplémentaires sont disponibles pour les langages de programmation suivants :

- Java

- Windows et .NET
- PHP
- Python
- Ruby

Pour plus d'informations sur les autres langages, consultezExemples de code et bibliothèques.

Applications de dépannage

MemoryDB fournit des erreurs spécifiques et descriptives pour vous aider à résoudre vos problèmes tout en interagissant avec l'API MemoryDB.

Récupération d'erreurs

Généralement, vous souhaitez que votre application vérifie si une demande a généré une erreur avant de passer du temps à traiter les résultats. Le moyen le plus simple de déterminer si une erreur s'est produite est de rechercher un`Error`dans la réponse de l'API MemoryDB.

La syntaxe XPath fournit une méthode simple pour rechercher la présence d'un nœud `Error` et récupérer le code et le message d'erreur. L'extrait de code suivant utilise Perl et le module XML::XPath pour déterminer si une erreur s'est produite lors d'une demande. Si une erreur s'est produite, le code imprime le premier code et message d'erreur dans la réponse.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Conseils pour le dépannage

Nous recommandons les processus suivants pour diagnostiquer et résoudre les problèmes avec l'API MemoryDB.

- Vérifiez que MemoryDB s'exécute correctement.

  Pour ce faire, il vous suffit d'ouvrir une fenêtre de navigateur et d'envoyer une demande de requête au service MemoryDB (par exemple, https://memory-db.us-east-1.amazonaws.com). Une

exception MissingAuthenticationTokenException ou UnknowNoPerationException confirme que le service est disponible et qu'il répond aux demandes.

- Vérifiez la structure de votre demande.

  Chaque opération MemoryDB possède une page de référence dans laRéférence d'API MemoryDB. Revérifiez que vous utilisez les paramètres correctement. Pour vous donner une idée des problèmes éventuels, observez les exemples de demandes ou de scénarios utilisateur pour voir s'ils effectuent des opérations similaires.

- Vérifiez le forum.

  MemoryDB possède un forum de discussion dans lequel vous pouvez chercher des solutions aux problèmes rencontrés par d'autres. Pour consulter le forum, rendez-vous à l'adresse

  https://forums.aws.amazon.com/.

Quotas pour MemoryDB

Votre AWS compte dispose de quotas par défaut, anciennement appelés limites, pour chaque AWS service. Sauf indication contraire, chaque quota est spécifique à la région. Vous pouvez demander des augmentations pour certains quotas, et d'autres quotas ne peuvent pas être augmentés.

Pour demander une augmentation de quota, consultez Demander une augmentation de quota dans le Guide de l'utilisateur de Service Quotas. Si le quota n'est pas encore disponible dans Service Quotas, utilisez le Formulaire d'augmentation de limite de service.

Votre AWS compte possède les quotas suivants liés à MemoryDB.

Ressource	Par défaut
Nœuds par région	300
Nœuds par cluster par type d'instance	90
Nœuds par partition	6
Groupes de paramètres par région	150
Groupes de sous-réseaux par région	150
Sous-réseaux par groupe de sous-réseaux	20
Utilisateurs par groupe d'utilisateurs	100
Nombre total d'utilisateurs	1 000
Nombre de groupes d'utilisateurs	100

Historique du document pour le guide de l'utilisateur de MemoryDB

Le tableau suivant décrit les versions de documentation de MemoryDB.

Modification	Description	Date
MemoryDB prend désormais en charge l'authentification des utilisateurs à l'aide d'IAM	L'authentification IAM vous permet d'authentifier une connexion à MemoryDB à l'aide d'identités. AWS Identity and Access Management Cela vous permet de renforcer votre modèle de sécurité et de simplifier de nombreuse s tâches administratives de sécurité. Pour plus d'informa tions, consultez Authenticating with IAM (Authentification avec IAM).	10 mai 2023
MemoryDB prend désormais en charge Redis OSS 7	Cette version apporte plusieurs nouvelles fonctionn alités à MemoryDB : fonctions Redis OSS, améliorations de l'ACL, Sharded Pub/Sub et multiplexage d'E/S amélioré. Pour plus d'informations, consultez la section Versions du moteur Redis OSS.	9 mai 2023
MemoryDB propose désormais des nœuds réservés	Les nœuds réservés vous offrent une réduction significa tive par rapport à la tarificat ion des nœuds à la demande. Les nœuds réservés ne sont	27 décembre 2022

	pas des nœuds physiques, mais plutôt une réduction de facturation appliquée à l'utilisation de nœuds à la demande dans votre compte. Pour plus d'informations, consultez la section Nœuds réservés de MemoryDB.	
MemoryDB prend désormais en charge la hiérarchisation des données	Hiérarchisation des données MemoryDB. Vous pouvez utiliser la hiérarchisation des données comme moyen moins coûteux de mettre à l'échelle vos clusters jusqu'à des centaines de téraoctets de capacité. Pour plus d'informations, consultez la rubrique Hiérarchisation des données.	3 novembre 2022

MemoryDB prend désormais en charge le format natif JSON (JavaScript Object Notation)	Le format natif JSON (JavaScript Object Notation) est un moyen simple et sans schéma d'encoder des ensembles de données complexes au sein de clusters Redis OSS. Vous pouvez stocker et accéder aux données de manière native à l'aide du format JSON (JavaScript Object Notation) dans les clusters Redis OSS et mettre à jour les données JSON stockées dans ces clusters, sans avoir à gérer de code personnalisé pour le sérialiser et le désérialiser. Pour plus d'informations, consultez Mise en route avec JSON.	25 mai 2022
MemoryDB prend désormais en charge AWS PrivateLink	AWS PrivateLink vous permet d'accéder en privé aux opérations de l'API MemoryDB sans passerelle Internet, périphérique NAT, connexion VPN ou connexion Direct AWS Connect. Pour plus d'informations, consultez les sections API MemoryDB et interface VPC endpoints ().AWS PrivateLink	24 janvier 2022

Première version	Première publication du guide de l'utilisateur de MemoryDB. Pour plus d'informations, voir Qu'est-ce que MemoryDB ?	19 août 2021

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.