Guide de l'utilisateur

Amazon ElastiCache

Version de l'API 2015-02-02

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ElastiCache: Guide de l’utilisateur

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d’Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n’est pas d’Amazon, d’une manière susceptible de créer une confusion parmi les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d’Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce que c'est ElastiCache ? ... 1
Mise en cache sans serveur ... 1
Clusters auto-conçus ... 2
Services connexes ... 2
Comment ça marche ... 3
Cache et moteurs de mise en cache ... 4
Choix entre deux options de déploiement ... 10
ElastiCache ressources ... 17
AWS Régions et zones de disponibilité ... 19
Cas d'utilisation ... 21
Commencer avec ElastiCache ... 29
Con ElastiCachefiguration ... 29
Inscrivez-vous pour un Compte AWS ... 29
Création d'un utilisateur doté d'un accès administratif ... 30
Octroi d'un accès par programmation ... 31
Configuration d'autorisations ... 33
Configurez EC2 ... 34
Octroi d'un accès au réseau ... 34
Configuration de l'accès par ligne de commande ... 36
Création d'un cache sans serveur Valkey ... 36
Lire et écrire des données ... 38
Nettoyage ... 40
Étapes suivantes ... 41
Création d'un cache sans serveur Redis OSS ... 41
Lire et écrire des données ... 42
Nettoyage ... 44
Étapes suivantes ... 45
Création d'un cache sans serveur Memcached ... 45
Lire et écrire des données ... 47
Nettoyage ... 52
Étapes suivantes ... 53
Tutoriels : Débuter avec Python et ElastiCache ... 53
Python et ElastiCache ... 54
Tutoriel : Configuration de Lambda pour un accès ElastiCache dans un VPC ... 72

Étape 1 : Création d'un cache ElastiCache sans serveur. ... 72
Étape 2 : créer une fonction Lambda pour ElastiCache ... 75
Étape 3 : tester la fonction Lambda avec ElastiCache ... 79
Étape 4 : Nettoyage (facultatif) ... 80
Conception de votre propre ElastiCache cluster ... 82
Composants et fonctionnalités ... 82
Nœuds ... 83
ElastiCache éclats ... 84
ElastiCache clusters ... 84
ElastiCache réplication ... 87
ElastiCache points de terminaison ... 89
Groupes de paramètres ... 90
ElastiCache sécurité ... 91
Groupes de sous-réseaux ... 91
ElastiCache sauvegardes ... 91
Événements ... 92
ElastiCache terminologie ... 93
Tutoriel : Comment concevoir votre propre cluster ... 95
Conception de votre propre ElastiCache cluster (Valkey) ... 95
Conception de votre propre cluster ElastiCache Redis OSS ... 117
Suppression d'un cluster ... 139
Autres tutoriels et vidéos ... 141
Vidéos ... 142
Gestion des nœuds dans ElastiCache ... 147
Affichage de l'état du ElastiCache nœud ... 148
Nœuds et partitions Valkey ou Redis OSS ... 154
Connexion aux nœuds ... 156
Types de nœuds pris en charge ... 161
Redémarrage des nœuds ... 176
Remplacement de nœuds (Valkey et Redis OSS) ... 181
Remplacement de nœuds (Memcached) ... 188
Nœuds réservés ... 190
Migration des nœuds de génération précédente ... 206
Gestion des clusters dans ElastiCache ... 210
Choix d'un type de réseau dans ElastiCache ... 212
Découverte automatique (Memcached) ... 217

Hiérarchisation des données ElastiCache ... 261
Préparation d'un cluster dans ElastiCache ... 268
Création d'un cluster pour Valkey ou Redis OSS ... 280
Création d'un cluster pour Memcached ... 290
Afficher les détails d'un ElastiCache cluster ... 294
Modification d'un ElastiCache cluster ... 308
Ajouter des nœuds à un ElastiCache cluster ... 315
Supprimer des nœuds d'un ElastiCache cluster ... 326
Annulation des opérations d'ajout ou de suppression de nœuds en attente dans ElastiCache ... 336
Supprimer un cluster dans ElastiCache ... 337
Accès à votre ElastiCache cluster ou à votre groupe de réplication ... 340
Recherche de points de terminaison de connexion dans ElastiCache ... 348
Des éclats dans ElastiCache ... 365
Comparaison des caches auto-conçus par Valkey, Memcached et Redis OSS ... 371
Migration en ligne pour Valkey ou Redis OSS ... 377
Présentation ... 377
Étapes de la migration ... 378
Préparation de votre source et de votre cible pour la migration ... 378
Test de la migration des données ... 381
Démarrage d'une migration ... 381
Vérification du déroulement de la migration des données ... 383
Achèvement de la migration des données ... 383
Effectuer la migration des données en ligne à l'aide de la console ... 384
Choix des régions et des zones de disponibilité pour ElastiCache ... 386
Considérations relatives à la zone de disponibilité avec Memcached ... 387
Localisation de vos nœuds ... 390
Régions et terminaux pris en charge ... 390
Utilisation de zones locales avec ElastiCache ... 396
Utiliser Outposts avec ElastiCache ... 397
Travailler avec ElastiCache ... 402
Instantané et restauration ... 402
Constraints ... 403
Impact sur les performances des sauvegardes de clusters auto-conçus ... 404
Planification des sauvegardes automatiques ... 406
Réalisation de sauvegardes manuelles ... 408

Création d'une sauvegarde finale ... 415
Description des sauvegardes ... 418
Copie de sauvegardes ... 420
Exportation d'une sauvegarde ... 422
Restaurer à partir d'une sauvegarde ... 431
Suppression d'une sauvegarde ... 434
Étiquetage des sauvegardes ... 436
Tutoriel : Création d'un cluster conçu par vos soins à l'aide d'une sauvegarde ... 437
Versions du moteur et mise à niveau dans ElastiCache ... 447
Mise à niveau des versions de moteur ... 447
Support étendu ... 452
Gestion des versions pour ElastiCache ... 455
Vallée 8.1 ... 462
Valkey 8.0 ... 462
ElastiCache version 7.2.6 pour Valkey ... 463
Versions de Redis OSS prises en charge ... 464
Calendrier de fin de vie des versions Redis OSS ... 478
Versions de Memcached prises en charge ... 480
Principales différences de comportement entre les versions du moteur et de compatibilité avec Valkey ... 485
Principales différences de comportement entre les versions du moteur et de compatibilité avec Redis OSS ... 485
Considérations en matière de mise à niveau lorsque vous utilisez des clusters auto-conçus ... 490
Bonnes pratiques et stratégies de mise en cache ... 492
Bonnes pratiques générales ... 493
Bonnes pratiques d'utilisation des répliques de lecture ... 494
Commandes Valkey, Memcached et Redis OSS prises en charge et restreintes ... 498
Configuration et limites de Valkey et Redis OSS ... 536
IPv6 exemples de clients pour Valkey, Memcached et Redis OSS ... 540
Bonnes pratiques pour les clients (Valkey et Redis OSS) ... 541
Bonnes pratiques pour les clients (Memcached) ... 566
Clusters à double pile ElastiCache compatibles TLS ... 570
Gestion de la mémoire réservée pour Valkey et Redis OSS ... 573
Meilleures pratiques lors de l'utilisation de clusters conçus par Valkey et Redis OSS ... 580
Stratégies de mise en cache pour Memcached ... 587

Gestion de votre cluster conçu par vos soins dans ElastiCache 593
Clusters Auto Scaling Valkey et Redis OSS 593
Modification du mode cluster 643
Réplication entre AWS régions à l'aide de banques de données mondiales 646
Haute disponibilité avec les groupes de réplication 676
Gestion de la maintenance des ElastiCache clusters 767
Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres 770
Connexion automatique d'une EC2 instance et d'un ElastiCache cache 887
EC2 Connectivité automatique 888
Affichage des ressources de calcul connectées 894
Dimensionnement ElastiCache 895
Mise à l'échelle ElastiCache sans serveur 895
Définition de limites de mise à l'échelle pour gérer les coûts 895
Pré-dimensionnement avec Serverless ElastiCache 895
Définition des limites de mise à l'échelle à l'aide de la console et AWS CLI 897
Mise à l'échelle de clusters conçus par vous-même 899
Commencer à utiliser les filtres Bloom 989
Type de données Bloom 990
Limite de taille des fleurs 991
ACLs 991
Métriques liées à Bloom 992
Commandes du filtre Bloom 993
Commencer à utiliser JSON pour Valkey et Redis OSS 994
Vue d'ensemble des types de données JSON 995
Commandes JSON 1007
Marquer vos ressources ElastiCache 1051
Surveillance des coûts avec des balises 1063
Gestion des balises à l'aide du AWS CLI 1065
Gestion des balises à l'aide de l' ElastiCache API 1070
Objectif Amazon ElastiCache Well-Architected 1073
Pilier Excellence opérationnelle 1074
Pilier Sécurité 1083
Reliability Pillar 1090
Pilier Efficacité des performances 1096
Cost Optimization Pillar 1108
Résolution des problèmes dans ElastiCache 1115

Problèmes de connexion ... 1115
Erreurs du client Valkey ou Redis OSS ... 1116
Résolution des problèmes de latence élevée en mode ElastiCache Serverless ... 1117
Résolution des problèmes de régulation dans Serverless ElastiCache ... 1119
Problèmes de connexion persistants ... 1120
Rubriques connexes ... 1142
Sécurité ... 1143
Protection des données ... 1144
Sécurité des données sur Amazon ElastiCache ... 1145
Confidentialité du trafic inter-réseau ... 1225
Amazon VPCs et la ElastiCache sécurité ... 1226
ElastiCache Points de terminaison VPC de l'API et de l'interface ()AWS PrivateLink ... 1251
Sous-réseaux et groupes de sous-réseaux ... 1257
Gestion de l'identité et des accès ... 1266
Public ciblé ... 1266
Authentification par des identités ... 1267
Gestion des accès à l'aide de politiques ... 1271
Comment Amazon ElastiCache travaille avec IAM ... 1274
Exemples de politiques basées sur l'identité ... 1281
Résolution des problèmes ... 1284
Contrôle d'accès ... 1286
Présentation de la gestion des accès ... 1288
Validation de conformité ... 1340
En savoir plus ... 1342
Résilience ... 1342
Atténuation des défaillances ... 1343
Sécurité de l'infrastructure ... 1348
Mises à jour de service ... 1349
Gestion des mises à jour des services ... 1349
Failles de sécurité corrigées ... 1354
Journalisation et surveillance ... 1357
Métriques et événements sans serveur pour Valkey et Redis OSS ... 1357
Métriques pour les caches sans serveur ... 1357
Événements pour les caches sans serveur ... 1368
Mesures et événements de cluster conçus par vos soins pour Valkey et Redis OSS ... 1384
Métriques pour les clusters auto-conçus ... 1384

Événements de cluster conçus par vos soins (Valkey et Redis OSS) 1384
Métriques et événements pour Memcached 1393
Métriques sans serveur Memcached 1393
Événements sans serveur Memcached 1397
Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail 1409
ElastiCache Informations Amazon dans CloudTrail 1409
Comprendre les entrées des fichiers ElastiCache journaux Amazon 1410
Surveillance des événements avec Amazon SNS 1414
Gestion des ElastiCache notifications Amazon SNS 1414
Affichage des ElastiCache événements 1420
Notifications d'événements Amazon SNS 1424
Livraison de journaux 1433
Contenu d'une entrée de journal lente 1434
Contenu d'une entrée dans le journal du moteur 1434
Autorisations pour configurer la journalisation 1434
Spécifications du type de journal et du format de journal 1435
ElastiCache destinations de journalisation 1436
Spécification de la diffusion des journaux à l'aide de la console 1439
Spécifier la livraison du journal à l'aide du AWS CLI 1440
Utilisation de la surveillance 1445
Métriques au niveau de l'hôte 1446
Métriques pour Valkey et Redis OSS 1450
Métriques pour Memcached 1469
Quelles métriques dois-je surveiller ? 1474
Choix des périodes et des statistiques de métriques 1479
Surveillance des métriques des CloudWatch clusters et des nœuds 1479
Quotas 1483
Référence 1486
Utilisation de l' ElastiCache API 1486
Utilisation de l'API Query 1486
Bibliothèques disponibles 1490
Applications de dépannage 1490
Configurez le AWS CLI pour ElastiCache 1492
Prérequis 1492
Téléchargement des outils de ligne de commande 1494
Installation des outils 1494

Indication des informations d'identification pour les outils 1495
Variables d'environnement 1496
Messages d'erreur 1497
Notifications 1499
ElastiCache Notifications générales 1499
ElastiCache pour les notifications Memcached 1499
ElastiCache pour les notifications spécifiques à Redis OSS 1500
ElastiCache Historique de la documentation 1501
AWS Glossaire 1544
..... mdxlv

Qu'est-ce qu'Amazon ElastiCache ?

Bienvenue dans le guide de ElastiCache l'utilisateur Amazon. Amazon ElastiCache est un service Web qui facilite la configuration, la gestion et le dimensionnement d'un stockage de données distribué en mémoire ou d'un environnement de cache dans le cloud. Il fournit une solution de cache performante, évolutive et économique. Dans le même temps, il permet de supprimer la complexité associée au déploiement et à la gestion d'un environnement de cache distribué.

Vous pouvez utiliser Amazon ElastiCache sous deux formats. Vous pouvez commencer avec un cache sans serveur ou choisir de concevoir votre propre cluster de cache.

> **Note**
>
> Amazon ElastiCache fonctionne avec les moteurs Valkey, Memcached et Redis OSS. Si vous n'êtes pas sûr du moteur que vous souhaitez utiliser, consultez Comparaison des caches auto-conçus par Valkey, Memcached et Redis OSS dans ce guide.

Mise en cache sans serveur

ElastiCache propose une mise en cache sans serveur, qui simplifie l'ajout et l'exploitation d'un cache pour votre application. ElastiCache Le mode Serverless vous permet de créer un cache hautement disponible en moins d'une minute et élimine le besoin de provisionner des instances ou de configurer des nœuds ou des clusters. Les développeurs peuvent créer un cache sans serveur en spécifiant le nom du cache à l'aide de la ElastiCache console, du SDK ou de la CLI.

ElastiCache Le mode Serverless élimine également le besoin de planifier et de gérer la capacité de mise en cache. ElastiCache surveille en permanence la mémoire cache, le calcul et la bande passante réseau utilisés par votre application, et évolue pour répondre aux besoins de votre application. ElastiCache offre une expérience de point de terminaison simple aux développeurs, en faisant abstraction de l'infrastructure de cache sous-jacente et de la conception du cluster. ElastiCache gère le provisionnement du matériel, la surveillance, le remplacement des nœuds et l'application des correctifs logiciels de manière automatique et transparente, afin que vous puissiez vous concentrer sur le développement des applications plutôt que sur l'exploitation du cache.

ElastiCache Serverless est compatible avec Valkey 7.2, Memcached 1.6.21 et versions ultérieures, et Redis OSS 7.1 et versions ultérieures.

Conception de votre propre ElastiCache cluster

Si vous avez besoin d'un contrôle précis de votre ElastiCache cluster, vous pouvez choisir de concevoir votre propre cluster Valkey, Memcached ou Redis OSS avec. ElastiCache ElastiCache vous permet de concevoir votre cluster en choisissant le type de nœud, le nombre de nœuds et le placement des nœuds dans les zones de AWS disponibilité de votre cluster. Comme il ElastiCache s'agit d'un service entièrement géré, il gère automatiquement le provisionnement du matériel, la surveillance, le remplacement des nœuds et les correctifs logiciels pour votre cluster.

La conception de votre propre ElastiCache cluster offre une flexibilité et un contrôle accrus sur vos clusters. Par exemple, vous pouvez choisir d'exploiter un cluster dans une seule zone de disponibilité ou plusieurs zones de disponibilité, en fonction de vos besoins. Vous pouvez également choisir d'exécuter Valkey, Memcached ou Redis OSS en mode cluster pour activer la mise à l'échelle horizontale, ou sans mode cluster pour une mise à l'échelle verticale uniquement. Lorsque vous concevez vos propres clusters, il vous incombe de choisir correctement le type et le nombre de nœuds afin de garantir que votre cache dispose de la capacité requise par votre application. Vous pouvez également choisir quand appliquer de nouveaux correctifs logiciels à votre cluster Valkey ou Redis OSS.

Lorsque vous concevez votre propre ElastiCache cluster, vous pouvez choisir d'exécuter Valkey 7.2 et versions ultérieures, Memcached 1.4 et versions ultérieures, ou Redis OSS 4.0 à 7.1 et versions ultérieures.

Services connexes

Base de données de mémoire

Lorsque vous décidez d'utiliser ElastiCache ou MemoryDB, considérez les comparaisons suivantes :

- ElastiCache 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 Valkey, Memcached ou Redis OSS. Vous devez envisager de mettre en cache ElastiCache 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 les cas d'utilisation dans lesquels vous souhaitez utiliser les structures de données Valkey ou Redis OSS et accéder APIs aux données stockées dans une base de données ou un magasin de données principal.

- ElastiCache peut également vous aider à réduire les coûts de base de données en stockant les données fréquemment consultées dans un cache. Si votre application a des exigences de débit de lecture élevées, vous pouvez atteindre une évolutivité élevée, des performances rapides et des coûts de stockage de données réduits en utilisant ElastiCache, au lieu de dimensionner, votre base de données sous-jacente.
- MemoryDB est une base de données en mémoire durable pour les charges de travail nécessitant une base de données principale ultrarapide. Il est compatible avec Valkey et Redis OSS. 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 Valkey ou Redis OSS et APIs 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.

Amazon Relational Database Service

ElastiCache peut vous aider à réduire les coûts de base de données en stockant les données fréquemment consultées dans un cache. Si votre application a des exigences de débit de lecture élevées, vous pouvez atteindre une évolutivité élevée, des performances rapides et des coûts de stockage de données réduits en utilisant ElastiCache, au lieu de dimensionner, votre base de données sous-jacente.

Pour plus d'informations générales sur le service Amazon Relational Database Service associé, consultez Amazon RDS

ElastiCache peut vous aider à réduire les coûts de base de données en stockant les données fréquemment consultées dans un cache. Si votre application a des exigences de débit de lecture élevées, vous pouvez atteindre une évolutivité élevée, des performances rapides et des coûts de stockage de données réduits en utilisant ElastiCache, au lieu de dimensionner, votre base de données sous-jacente.

Comment ElastiCache fonctionne

Vous trouverez ici un aperçu des principaux composants d'un ElastiCache déploiement.

Cache et moteurs de mise en cache

Un cache est un magasin de données en mémoire que vous pouvez utiliser pour stocker des données mises en cache. En général, votre application met en cache les données fréquemment consultées dans un cache afin d'optimiser les temps de réponse. ElastiCache propose deux options de déploiement : des clusters sans serveur et des clusters conçus par vos soins. Consultez Choix entre deux options de déploiement.

> **Note**
>
> Amazon ElastiCache fonctionne avec les moteurs Valkey, Memcached et Redis OSS. Si vous n'êtes pas sûr du moteur que vous souhaitez utiliser, consultez Comparaison des caches auto-conçus par Valkey, Memcached et Redis OSS dans ce guide.

Rubriques

- Comment ElastiCache fonctionne
- Grille tarifaire
- ElastiCache sauvegardes

Comment ElastiCache fonctionne

ElastiCache Serverless (Sans serveur)

ElastiCache Le mode Serverless vous permet de créer un cache sans vous soucier de la planification des capacités, de la gestion du matériel ou de la conception du cluster. Il vous suffit de donner un nom à votre cache et vous recevez un point de terminaison unique que vous pouvez configurer dans votre client Valkey, Memcached ou Redis OSS pour commencer à accéder à votre cache.

> **Note**
>
> - ElastiCache Serverless exécute Valkey, Memcached ou Redis OSS en mode cluster et n'est compatible qu'avec les clients prenant en charge le protocole TLS.

Principaux avantages

- Aucune planification de la capacité : ElastiCache sans serveur, vous n'avez plus besoin de planifier la capacité. ElastiCache Serverless surveille en permanence l'utilisation de la mémoire, du calcul et de la bande passante réseau de votre cache et évolue à la fois verticalement et horizontalement. Il permet à un nœud de cache d'augmenter de taille, tout en lançant en parallèle une opération de montée en puissance afin de garantir que le cache puisse être mis à l'échelle conformément aux exigences de votre application à tout moment.
- P ay-per-use : Avec ElastiCache Serverless, vous payez pour les données stockées et le calcul utilisé par votre charge de travail dans le cache. Consultez Grille tarifaire.
- Haute disponibilité : ElastiCache Serverless réplique automatiquement vos données sur plusieurs zones de disponibilité (AZ) pour une haute disponibilité. Il surveille automatiquement les nœuds de cache sous-jacents et les remplace en cas de défaillance. Il offre un niveau de service (SLA) de disponibilité de 99,99 % pour chaque cache.
- Mises à niveau logicielles automatiques : ElastiCache Serverless met automatiquement à niveau votre cache vers la dernière version mineure et corrige le logiciel sans aucun impact sur la disponibilité de votre application. Lorsqu'une nouvelle version majeure sera disponible, je vous ElastiCache enverrai une notification.
- Sécurité : ElastiCache sans serveur chiffre toujours les données en transit et au repos. Vous pouvez utiliser une clé gérée par un service ou votre propre clé gérée par le client pour chiffrer les données au repos.

Le schéma suivant illustre le fonctionnement de ElastiCache Serverless.

Lorsque vous créez un nouveau cache sans serveur, vous ElastiCache créez un point de terminaison Virtual Private Cloud (VPC) dans les sous-réseaux de votre choix dans votre VPC. Votre application peut se connecter au cache via ces points de terminaison de VPC.

Avec ElastiCache Serverless, vous recevez un point de terminaison DNS unique auquel votre application se connecte. Lorsque vous demandez une nouvelle connexion au point de terminaison, ElastiCache Serverless gère toutes les connexions au cache via une couche proxy. La couche proxy permet de réduire la complexité de la configuration du client, car le client n'a pas besoin de redécouvrir la topologie du cluster en cas de modification du cluster sous-jacent. La couche proxy est un ensemble de nœuds proxy qui gèrent les connexions à l'aide d'un équilibreur de charge de réseau.

Lorsque votre application crée une connexion au cache, la demande est envoyée à un nœud proxy par l'équilibreur de charge de réseau. Lorsque votre application exécute des commandes de cache, le nœud proxy connecté à votre application exécute les demandes sur un nœud de cache de votre cache. La couche proxy extrait la topologie et les nœuds du cluster de cache de votre client. Cela permet d' ElastiCache équilibrer intelligemment la charge, d'étendre et d'ajouter de nouveaux nœuds de cache, de remplacer les nœuds de cache en cas de défaillance et de mettre à jour le logiciel sur les nœuds de cache, le tout sans impact sur la disponibilité de votre application ou sans avoir à réinitialiser les connexions.

Clusters conçus ElastiCache par vos soins

Vous pouvez choisir de concevoir vos propres ElastiCache clusters en choisissant une famille de nœuds de cache, une taille et un nombre de nœuds pour votre cluster. La conception de votre propre cluster vous offre un contrôle plus précis et vous permet de choisir le nombre de partitions de votre cache et le nombre de nœuds (primaires et de réplica) dans chaque partition. Vous pouvez choisir d'utiliser Valkey ou Redis OSS en mode cluster en créant un cluster avec plusieurs partitions, ou en mode non cluster avec une seule partition.

Principaux avantages

- Concevez votre propre cluster : avec ElastiCache, vous pouvez concevoir votre propre cluster et choisir où vous souhaitez placer vos nœuds de cache. Par exemple, si vous souhaitez trouver un compromis entre haute disponibilité et faible latence pour votre application, vous pouvez choisir de déployer vos nœuds de cache dans une seule zone de disponibilité. Vous pouvez également concevoir votre cluster avec des nœuds répartis sur plusieurs AZs pour atteindre une haute disponibilité.
- Contrôle précis : lorsque vous concevez votre propre cluster, vous pouvez mieux contrôler le réglage précis des paramètres de votre cache. Par exemple, vous pouvez utiliser Paramètres OSS Valkey et Redis ou Paramètres spécifiques Memcached configurer le moteur de cache.
- Mise à l'échelle verticale et horizontale : vous pouvez choisir de mettre à l'échelle manuellement votre cluster en augmentant ou en diminuant la taille du nœud de cache selon vos besoins. Vous pouvez également effectuer une mise à l'échelle horizontale en ajoutant de nouvelles partitions ou d'autres réplicas à vos partitions. Vous pouvez également utiliser la fonction Auto-Scaling pour configurer le dimensionnement en fonction d'un calendrier ou en fonction de métriques telles que l'utilisation du processeur et de la mémoire dans le cache.

Le schéma suivant illustre le fonctionnement des clusters ElastiCache auto-conçus.

Grille tarifaire

Vous pouvez effectuer ElastiCache le déploiement selon deux options de déploiement. Lorsque vous déployez ElastiCache Serverless, vous payez l'utilisation des données stockées en Go par heure et le calcul en unités de ElastiCache traitement (ECPU). Lorsque vous choisissez de concevoir vos propres ElastiCache clusters, vous payez par heure d'utilisation du nœud de cache. Consultez les détails de la tarification en cliquant ici.

Stockage de données

Vous payez les données stockées dans ElastiCache Serverless, facturées en gigaoctets (GB-heures). ElastiCache Serverless surveille en permanence les données stockées dans votre cache, en échantillonnant plusieurs fois par minute, et calcule une moyenne horaire pour déterminer l'utilisation du stockage de données du cache en Go d'heures. Chaque cache ElastiCache sans serveur est mesuré pour un minimum de 1 Go de données stockées.

ElastiCache Unités de traitement (ECPUs)

Vous payez les requêtes que votre application exécute sur ElastiCache Serverless dans ElastiCache Processing Units (ECPUs), une unité qui inclut à la fois le temps de vCPU et les données transférées.

- Les lectures et écritures simples exigent 1 ECPU pour chaque kilo-octet (Ko) de données transféré. Par exemple, une commande GET qui transfère jusqu'à 1 Ko de données consomme 1 ECPU. Une requête SET qui transfère 3,2 Ko de données en consommera 3,2 ECPUs.
- Avec Valkey et Redis OSS, les commandes qui consomment plus de temps de vCPU et transfèrent plus de données consomment en ECPUs fonction de la plus élevée des deux dimensions. Par exemple, si votre application utilise la commande HMGET, consomme 3 fois plus de temps vCPU qu'une SET/GET simple commande et transfère 3,2 Ko de données, elle consommera 3,2 ECPU. Sinon, s'il ne transfère que 2 Ko de données, il en consommera 3 ECPUs.
- Avec Valkey et Redis OSS, les commandes qui nécessitent un temps de vCPU supplémentaire consommeront proportionnellement plus de temps. ECPUs Par exemple, si votre application utilise la commande Valkey ou Redis OSS HMGET et consomme 3 fois plus de temps de vCPU qu'une simple SET/GET commande, elle en consommera 3. ECPUs
- Avec Memcached, les commandes qui agissent sur plusieurs éléments consommeront proportionnellement plus. ECPUs Par exemple, si votre application exécute un multiget sur 3 éléments, elle en consommera 3 ECPUs.
- Avec Memcached, les commandes qui agissent sur un plus grand nombre d'éléments et transfèrent plus de données consomment ECPUs en fonction de la plus élevée des deux dimensions. Par exemple, si votre application utilise la commande GET, récupère 3 éléments et transfère 3,2 Ko de données, elle consommera 3,2 ECPU. Sinon, s'il ne transfère que 2 Ko de données, il en consommera 3 ECPUs.

ElastiCache Serverless émet une nouvelle métrique appelée `ElastiCacheProcessingUnits` qui vous aide à comprendre la ECPUs consommation de votre charge de travail.

Heures d'utilisation des nœuds

Vous pouvez choisir de concevoir votre propre cluster de cache en choisissant la famille de EC2 nœuds, la taille, le nombre de nœuds et leur emplacement dans les zones de disponibilité. Lorsque vous concevez vous-même votre cluster, vous payez par heure pour chaque nœud de cache.

ElastiCache sauvegardes

Une sauvegarde est une point-in-time copie d'un cache sans serveur ou d'un cluster auto-conçu par Valkey ou Redis OSS. ElastiCache vous permet de sauvegarder vos données à tout moment ou de

configurer des sauvegardes automatiques. Les sauvegardes peuvent être utilisées pour restaurer un cache existant ou pour alimenter un nouveau cache. Les sauvegardes sont constituées de toutes les données d'un cache, plus quelques métadonnées. Pour de plus amples informations, veuillez consulter Instantané et restauration.

Choix entre deux options de déploiement

Amazon ElastiCache propose deux options de déploiement :

- Mise en cache sans serveur
- Clusters auto-conçus

Pour obtenir la liste des commandes prises en charge pour les deux, consultezCommandes Valkey, Memcached et Redis OSS prises en charge et restreintes.

Mise en cache sans serveur

Amazon ElastiCache Serverless simplifie la création de cache et s'adapte instantanément pour prendre en charge les applications les plus exigeantes des clients. Avec ElastiCache Serverless, vous pouvez créer un cache hautement disponible et évolutif en moins d'une minute, éliminant ainsi le besoin de provisionner, de planifier et de gérer la capacité du cluster de cache. ElastiCache Serverless stocke automatiquement les données de manière redondante dans trois zones de disponibilité et fournit un accord de niveau de service (SLA) de disponibilité de 99,99 %. Les sauvegardes provenant de clusters Valkey ou Redis OSS conçus par nos soins peuvent être restaurées dans une configuration sans serveur.

Clusters auto-conçus

Si vous avez besoin d'un contrôle précis de votre cluster Valkey, Memcached ou Redis OSS, vous pouvez choisir de concevoir votre propre cluster avec. ElastiCache ElastiCache vous permet d'exploiter un cluster basé sur des nœuds, en choisissant le type de nœud, le nombre de nœuds et le placement des nœuds dans les zones de AWS disponibilité de votre cluster. Comme il ElastiCache s'agit d'un service entièrement géré, il permet de gérer le provisionnement du matériel, la surveillance, le remplacement des nœuds et les correctifs logiciels pour votre cluster. Les clusters conçus par nos soins peuvent être conçus pour fournir un SLA de disponibilité allant jusqu'à 99,99 %. Les sauvegardes à partir de caches Valkey ou Redis OSS sans serveur peuvent être restaurées dans un cluster conçu par vos soins.

Choix entre deux options de déploiement

Choisissez la mise en cache sans serveur si :

- Vous créez un cache pour les charges de travail nouvelles ou difficiles à prévoir.
- Vous avez un trafic imprévisible au niveau de l'application.
- Vous cherchez la méthode la plus simple pour démarrer avec un cache.

Choisissez de concevoir votre propre ElastiCache cluster si :

- Vous utilisez déjà ElastiCache Serverless et souhaitez mieux contrôler le type de nœud exécutant Valkey, Memcached ou Redis OSS, le nombre de nœuds et le placement de ces nœuds.
- Vous vous attendez à ce que le trafic de vos applications soit relativement prévisible et souhaitez un contrôle précis des performances, de la disponibilité et des coûts.
- Vous pouvez prévoir vos exigences de capacité pour contrôler les coûts.

Comparaison entre la mise en cache sans serveur et les clusters conçus par vos soins

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Configuration du cache	Créez un cache avec juste un nom en moins d'une minute	Permet un contrôle précis de la conception des clusters de cache. L'utilisateur peut choisir le type de nœud, le nombre de nœuds et leur emplacement dans les zones de disponibilité AWS
ElastiCache Version prise en charge	Valkey 7.2 et supérieur, Redis OSS version 7.1 et supérieur, Memcached 1.6.21 et supérieur	Valkey 7.2 et supérieur, Redis OSS version 4.0 et supérieur, Memcached 1.4 et supérieur
Mode cluster (Valkey et Redis OSS)	Fait fonctionner les moteurs `cluster mode enabled` uniquement. Les clients doivent prendre en charge `cluster mode enabled`	Peut être configuré pour fonctionner en mode cluster activé ou en mode cluster désactivé.

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
	la connexion à ElastiCache Serverless.	
Mise à l'échelle	Adapte automatiquement les moteurs à la fois verticale ment et horizontalement sans aucune gestion de capacité.	Permet de contrôler la mise à l'échelle, tout en exigeant une surveillance pour s'assurer que la capacité actuelle répond correctement à la demande. Pour Valkey et Redis OSS, vous pouvez choisir d'effectu er une mise à l'échelle verticale en augmentant ou en diminuant la taille du nœud de cache selon les besoins. Vous pouvez également effectuer une mise à l'échelle horizonta le, en ajoutant de nouveaux fragments ou en ajoutant des répliques à vos fragments. Cette fonctionnalité n'est pas disponible pour Memcached. Grâce à la fonction Auto-Scal ing, vous pouvez également configurer le dimensionnement en fonction d'un calendrier ou en fonction de mesures telles que l'utilisation du processeu r et de la mémoire dans le cache.

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Connexion client	Les clients se connectent à un seul point de terminaison. Cela permet à la topologie du nœud de cache sous-jacent (dimensionnement, remplacements et mises à niveau) de changer sans déconnecter le client.	Les clients se connectent à chaque nœud de cache individuel. Si un nœud est remplacé, le client redécouvre la topologie du cluster et rétablit les connexions.
Configurabilité	Aucune configuration précise n'est disponible. Les clients peuvent configurer les paramètres de base, notamment les sous-réseaux qui peuvent accéder au cache, si les sauvegardes automatiques sont activées ou non, et les limites d'utilisation maximale du cache.	Les clusters conçus par nos soins fournissent des options de configuration précises. Les clients peuvent utiliser des groupes de paramètres pour un contrôle précis. Pour un tableau des valeurs de ces paramètres de type de nœud, consultez Paramètres spécifiques au moteur.
Multi-AZ	Les données sont répliquées de manière asynchrone sur plusieurs zones de disponibilité pour une meilleure disponibilité et une latence de lecture améliorée.	Permet de concevoir le cluster dans une seule zone de disponibilité ou dans plusieurs zones de disponibilité (AZs). Lorsque vous utilisez Valkey ou Redis OSS, fournit des clusters multi-AZ avec des données répliquées de manière asynchrone sur plusieurs zones de disponibilité pour une meilleure disponibilité et une latence de lecture améliorée.

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Chiffrement au repos	Toujours activé. Les clients peuvent utiliser une clé Clé gérée par AWS ou une clé gérée par le client dans AWS KMS.	Option permettant d'activer ou de désactiver le chiffrement au repos. Lorsque cette option est activée, les clients peuvent utiliser une clé Clé gérée par AWS ou une clé gérée par le client AWS KMS.
Chiffrement en transit (TLS)	Toujours activé. Les clients doivent prendre en charge la connectivité TLS.	Option permettant d'activer ou de désactiver.
Sauvegardes	Prend en charge les sauvegardes automatiques et manuelles des caches sans impact sur les performances. Les sauvegardes Valkey et Redis OSS sont compatibl es entre elles et peuvent être restaurées dans un cache ElastiCache sans serveur ou dans un cluster conçu par vos soins.	Supporte les sauvegardes automatiques et manuelles pour Valkey et Redis OSS. Les clusters peuvent avoir un impact sur les performan ces en fonction de la mémoire réservée disponible. Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS. Les sauvegardes Valkey et Redis OSS sont compatibl es entre elles et peuvent être restaurées dans un cache ElastiCache sans serveur ou dans un cluster conçu par vos soins.

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Surveillance	Support des mesures au niveau du cache, notamment le taux de réussite du cache, le taux d'échec du cache, la taille des données et la ECPUs consommation. ElastiCache Serverless envoie des événements EventBrid ge lorsque des événement s importants se produisen t dans votre cache. Vous pouvez choisir de surveiller, d'ingérer, de transformer les ElastiCache événements et d'agir en conséquence à l'aide d'Amazon EventBridge. Pour de plus amples informations, veuillez consulter Événements pour les caches sans serveur.	ElastiCache les clusters auto-conçus émettent des métriques au niveau de chaque nœud, y compris des métriques au niveau de l'hôte et des métriques du cache. Les clusters conçus par leurs soins émettent des notificat ions SNS pour les événement s importants. Consultez Métriques pour Memcached et Métriques pour Valkey et Redis OSS.
Disponibilité	Contrat de niveau de service (SLA) de disponibilité de 99,99 %	Les clusters conçus par nos soins peuvent être conçus pour atteindre un accord de niveau de service (SLA) de disponibilité allant jusqu'à 99,99 %, en fonction de la configuration.

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Mises à niveau logicielles et application de correctifs	Met automatiquement à niveau le logiciel de cache vers la dernière version mineure et corrective, sans impact sur les applications. Les clients reçoivent une notification pour les mises à niveau des versions majeures, et les clients peuvent passer à la dernière version majeure quand ils le souhaitent.	Les clusters conçus par le client offrent un libre-service adapté aux besoins des clients pour les mises à niveau mineures et les mises à niveau des versions comportant des correctifs, ainsi que pour les mises à niveau des versions majeures. Les mises à jour gérées sont automatiquement appliquées pendant les périodes de maintenance définies par le client. Les clients peuvent également choisir d'appliquer une mise à niveau de version mineure ou de correctif à la demande.
Magasin de données mondial	Non pris en charge	Supporte le Global Data Store, qui permet la réplication entre régions avec des écritures dans une seule région et des lectures multirégionales

Fonctionnalité	Mise en cache sans serveur	Clusters auto-conçus
Hiérarchisation des données	Non pris en charge	Les clusters conçus à l'aide de nœuds de la famille r6gd voient leurs données hiérarchisées entre la mémoire et le stockage SSD (Solid State Drive) local. La hiérarchisation des données constitue une option rentable pour les charges de travail Valkey et Redis OSS en utilisant des disques SSD (SSDs) à moindre coût dans chaque nœud du cluster, en plus du stockage des données en mémoire.
Modèle de tarification	Pay-per-use, sur la base des données stockées en Go d'heures et des demandes dans les unités de ElastiCache traitement (ECPU). Consultez les détails de la tarification en cliquant ici.	Pay-per-hour, en fonction de l'utilisation du nœud de cache. Consultez les détails de la tarification en cliquant ici.

Voir aussi:

- Conception et gestion de votre propre ElastiCache cluster

ElastiCache Ressources Amazon pour les nouveaux utilisateurs

Nous recommandons aux nouveaux utilisateurs de commencer par lire les sections suivantes et de les consulter au besoin.

- Points forts du service et prix — La page détaillée du produit fournit un aperçu général du produit ElastiCache, les points forts du service et les prix.

- ElastiCache vidéos — La ElastiCache Vidéos section contient des vidéos qui vous présentent Amazon ElastiCache. Les vidéos présentent des cas d'utilisation courants ElastiCache et montrent comment les utiliser ElastiCache pour réduire la latence et améliorer le débit de vos applications.
- Mise en route — la section Commencer à utiliser Amazon ElastiCache contient des informations sur la création d'un cluster de cache. Elle explique également comment autoriser l'accès au cluster de cache, se connecter au nœud du cache et supprimer le cluster de cache.
- Performance à grande échelle — Le ElastiCache livre blanc Performance at scale with Amazon aborde les stratégies de mise en cache qui permettent à votre application de fonctionner correctement à grande échelle.

Une fois que vous avez terminé les sections précédentes, lisez ces sections :

- Choix de la taille de votre nœud

  Vous souhaitez que vos nœuds soient suffisamment grands pour pouvoir accueillir toutes les données à mettre en cache. En même temps, vous ne souhaitez pas payer pour plus d'espace de cache que ce dont vous avez besoin. vous pouvez vous appuyer sur cette rubrique pour sélectionner la meilleure taille de nœud.
- ElastiCache meilleures pratiques et stratégies de mise en cache

  Elles permettent d'identifier et de résoudre les problèmes qui peuvent affecter l'efficacité de votre cluster.

Si vous souhaitez utiliser le AWS Command Line Interface (AWS CLI), vous pouvez utiliser les documents suivants pour vous aider à démarrer :

- AWS Command Line Interface documentation

  Cette section fournit des informations sur le téléchargement AWS CLI, le AWS CLI fonctionnement de votre système et la fourniture de vos AWS informations d'identification.
- AWS CLI documentation pour ElastiCache

  Ce document distinct couvre toutes les ElastiCache commandes AWS CLI for, y compris la syntaxe et les exemples.

Vous pouvez écrire des programmes d'application pour utiliser l' ElastiCache API avec une variété de langages de programmation courants. Voici quelques ressources :

- Outils pour Amazon Web Services

  Amazon Web Services fournit un certain nombre de kits de développement logiciel (SDKs) avec support pour ElastiCache. Vous pouvez coder pour ElastiCache utiliser Java, .NET, PHP, Ruby et d'autres langages. Ils SDKs peuvent considérablement simplifier le développement de votre application en formatant vos demandes, en analysant les réponses ElastiCache, en fournissant une logique de nouvelle tentative et en gérant les erreurs.

- Utilisation de l' ElastiCache API

  Si vous ne souhaitez pas utiliser le AWS SDKs, vous pouvez interagir ElastiCache directement avec l'API Query. Cette section fournit des astuces de dépannage et des informations pour créer et authentifier les requêtes et la façon de gérer les réponses.

- Référence ElastiCache d'API Amazon

  Ce document distinct couvre toutes les opérations de l' ElastiCache API, y compris la syntaxe et les exemples.

AWS Régions et zones de disponibilité

Les ressources du cloud computing Amazon sont hébergées dans des installations de centres de données hautement disponible de différentes régions du monde (par exemple, Amérique du Nord, Europe et Asie). Chaque emplacement de centre de données est appelé une AWS région.

Chaque AWS région contient plusieurs emplacements distincts appelés zones de disponibilité, ou AZs. Chaque zone de disponibilité est conçue pour être isolée des défaillances dans d'autres zones de disponibilité. Chacune est conçue pour fournir une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même AWS région. En lançant des instances dans des zones de disponibilité distinctes, vous pouvez protéger vos applications de la défaillance d'un seul emplacement. Pour plus d'informations, consultez la page Choix des régions et zones de disponibilité.

Vous pouvez créer votre cluster dans plusieurs zones de disponibilité, option appelée déploiement Multi-AZ. Lorsque vous choisissez cette option, Amazon effectue automatiquement la mise en service et la maintenance d'une instance de base de données de secours secondaire dans une autre zone de disponibilité. Votre instance de nœud primaire est répliquée de manière asynchrone entre les zones de disponibilité (AZ) dans l'instance secondaire. Cette approche permet de fournir une assistance en matière de redondance et de basculement des données, d'éliminer les I/O blocages

et de minimiser les pics de latence lors des sauvegardes du système. Pour plus d'informations, consultez Minimiser les temps d'arrêt ElastiCache pour Valkey et Redis OSS avec Multi-AZ.

Cas ElastiCache d'utilisation courants et comment ElastiCache vous pouvez vous y aider

Qu'il s'agisse de communiquer les dernières actualités, le classement des 10 meilleurs, un catalogue de produits, ou de vendre des billets d'entrée pour un événement, tout est une question de rapidité. Le succès de votre site web et de votre activité dépend essentiellement de la vitesse à laquelle vous diffusez du contenu.

Dans son article « For Impatient Web Users, an Eye Blink Is Just Too Long to Wait (pour les utilisateurs Web impatients, patienter le temps d'un clignement d'œil est encore trop long) », le New York Times note que les utilisateurs peuvent enregistrer une différence de 250 millisecondes (1/4 de seconde) entre des sites concurrents. Les utilisateurs ont tendance à éliminer le site le plus lent en faveur du site le plus rapide. Les tests effectués chez Amazon, cités dans l'article How Webpage Load Time Is Related to Visitor Loss (comment le délai de chargement d'une page Web est lié à la perte de visiteurs), a révélé que pour chaque augmentation du temps de chargement de 100 ms (1/10 de seconde), les ventes diminuent de 1 %.

Si quelqu'un recherche des données, vous pouvez fournir ces données beaucoup plus rapidement si elles sont mises en cache. Cela est vrai que ce soit pour une page web ou un rapport qui permet de prendre des décisions métier. Votre entreprise peut-elle se permettre de ne pas mettre en cache ses pages Web afin de les diffuser avec la plus brève latence possible ?

Intuitivement, il peut être évident de vouloir mettre en cache vos éléments les plus fortement demandés. Mais pourquoi ne pas mettre en cache vos éléments les moins fréquemment demandés ? Même la requête de base de données la plus optimisée ou l'appel d'API à distance va prendre considérablement plus de temps que la récupération d'une clé horizontale à partir d'un cache en mémoire. Sensiblement plus lent est ce qui a tendance à faire fuir les clients.

Les exemples suivants illustrent certaines des manières dont l'utilisation ElastiCache peut améliorer les performances globales de votre application.

Rubriques

- Stockage de données en mémoire
- Classements de jeu
- Messagerie (Pub/Sub)
- Données de recommandation (hachages)
- ElastiCache Témoignages de clients

Stockage de données en mémoire

Le principal objectif d'un magasin clé/valeur en mémoire est de fournir un accès ultra rapide (latence inférieure à la milliseconde) et un accès abordable aux copies de données. La plupart des magasins de données ont des zones de données qui sont fréquemment consultées mais rarement mises à jour. En outre, l'interrogation d'une base de données est toujours plus lente et moins chère que la recherche d'une clé dans le cache d'une paire clé-valeur. Certaines requêtes de base de données sont particulièrement onéreuses à effectuer. Par exemple, les requêtes qui impliquent des jointures entre plusieurs tables ou des requêtes avec des calculs intensifs. En mettant en cache les résultats de cette requête, vous ne payez le prix de la requête qu’une seule fois. Ensuite, vous pouvez récupérer rapidement les données plusieurs fois sans avoir à réexécuter la requête.

Que dois-je mettre en cache ?

Lorsque vous choisissez les données à mettre en cache, tenez compte des facteurs suivants :

Vitesse et coûts : il est toujours plus lent et plus cher d'acquérir des données à partir d'une base de données qu'à partir d'un cache. Certaines requêtes de base de données sont, par nature, plus lentes et plus chères que les autres. Par exemple, les requêtes qui effectuent des jointures entre plusieurs tables sont nettement plus lentes et plus onéreuses que les requêtes de table simples. Si les données intéressantes à acquérir nécessitent une requête lente et coûteuse, il serait judicieux de les mettre en cache. Si l'acquisition de données nécessite une requête relativement simple et rapide, elles peuvent toujours être mises en cache en fonction des autres facteurs.

Data and access pattern (Données et modèle d'accès) : déterminer les données à mettre en cache nécessite aussi de comprendre les données elles-mêmes ainsi que leurs modèles d'accès. À titre d'exemple, il n'est pas judicieux de mettre en cache des données qui changent rapidement ou qui sont rarement consultées. Pour que la mise en cache constitue un réel avantage, les données doivent être relativement statiques et fréquemment consultées. Par exemple, un profil personnel sur un site de médias sociaux. En revanche, vous ne voulez pas mettre en cache des données si la mise en cache ne constitue aucun avantage en matière de vitesse et de coût. Par exemple, il n'est pas logique de mettre en cache des pages web qui renvoient des résultats de recherche, car les requêtes et les résultats sont généralement uniques.

Importance : par définition, les données mises en cache sont des données obsolètes. Même si dans certains cas elles ne le sont pas, elles doivent toujours être considérées et traitées comme telles. Pour déterminer si vos données peuvent être mises en cache, déterminez la tolérance de votre application concernant les données obsolètes.

Votre application peut être en mesure de tolérer des données obsolètes dans un contexte, mais pas un autre. Par exemple, supposons que votre site sert un cours d'action coté en bourse. Vos clients peuvent accepter une certaine moralité avec une clause de non-responsabilité selon laquelle les prix peuvent être n minutes de retard. Cependant, si vous communiquez le cours pour la même action à un courtier effectuant une vente ou un achat, vous aurez besoin de données en temps réel.

Envisagez de mettre en cache vos données dans les cas suivants :

- elles sont trop lentes ou onéreuses à acquérir en comparaison à la récupération de cache.
- Les utilisateurs accèdent souvent à vos données.
- Vos données restent relativement les mêmes, ou si elles changent rapidement, le manque de stabilité n'est pas un gros problème.

Pour de plus amples informations, consultez Stratégies de mise en cache pour Memcached.

Classements de jeu

Avec les ensembles triés Valkey ou Redis OSS, vous pouvez déplacer la complexité informatique des classements de votre application vers votre cluster.

Des classements, tels que les 10 meilleurs pour un jeu, nécessitent des calculs complexes. Cela est particulièrement vrai lorsque le nombre de joueurs est important et que les scores changent constamment. Les ensembles triés Valkey et Redis OSS garantissent à la fois l'unicité et l'ordre des éléments. Avec les ensembles triés, chaque fois qu'un nouvel élément est ajouté à l'ensemble trié, il est reclassé en temps réel. Cela est ensuite ajouté à l'ensemble dans son ordre numérique approprié.

Dans le schéma suivant, vous pouvez voir comment fonctionne un classement de ElastiCache jeu.

Example Classement de Valkey ou Redis OSS

Dans cet exemple, quatre joueurs et leurs résultats sont saisis dans une liste triée à l'aide de `ZADD`. La commande `ZREVRANGEBYSCORE` répertorie les joueurs en fonction de leur score, par ordre décroissant. Ensuite, la commande `ZADD` est utilisée pour mettre à jour le score de Jeanne en remplaçant l'entrée existante. Enfin, `ZREVRANGEBYSCORE` répertorie les joueurs en fonction de leur score, par ordre décroissant. La liste montre que Jeanne a grimpé dans les classements.

```
ZADD leaderboard 132 Robert
ZADD leaderboard 231 Sandra
ZADD leaderboard 32 June
ZADD leaderboard 381 Adam

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) Sandra
3) Robert
4) June

ZADD leaderboard 232 June

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) June
```

```
3) Sandra
4) Robert
```

La commande suivante permet à Jeanne de voir où elle se classe parmi tous les joueurs. Comme le classement part de zéro, ZREVRANK renvoie un 1 pour June, ce qui la place en deuxième position.

```
ZREVRANK leaderboard June
1
```

Pour plus d'informations, consultez la documentation de Valkey sur les ensembles triés.

Messagerie (Pub/Sub)

Lorsque vous envoyez un message électronique, vous envoyez à un ou plusieurs destinataires spécifiés. Dans le paradigme pub/sub de Valkey et Redis OSS, vous envoyez un message à un canal spécifique sans savoir qui, le cas échéant, le reçoit. Les personnes qui reçoivent le message sont celles qui sont abonnées à la chaîne. Par exemple, supposons que vous vous abonniez à la chaîne news.sports.golf. Vous et toutes les autres personnes abonnées à la chaîne news.sports.golf reçoivent des messages publiés dans news.sports.golf.

La fonctionnalité PUB/Sub n'a aucun rapport avec un espace clé. Par conséquent, elle n'interfère à aucun niveau. Dans le schéma suivant, vous trouverez une illustration de la ElastiCache messagerie avec Valkey et Redis OSS.

Abonnement en cours

Pour recevoir des messages sur une chaîne, vous devez vous y être abonné. Vous pouvez vous abonner à une seule chaîne, à plusieurs chaînes spécifiées ou à toutes les chaînes qui correspondent à un modèle. Pour annuler un abonnement, vous devez vous désinscrire de la chaîne spécifiée lors de l'inscription. Ou, si vous vous êtes abonné à l'aide de la mise en correspondance de modèle, vous vous désabonnez en utilisant le même modèle que celui utilisé précédemment.

Example – Abonnement à une chaîne unique

Pour vous inscrire à une seule chaîne, utilisez la commande SUBSCRIBE en spécifiant la chaîne à laquelle vous voulez vous abonner. Dans l'exemple suivant, un client s'abonne à la chaîne news.sports.golf.

```
SUBSCRIBE news.sports.golf
```

Après un certain temps, le client annule son abonnement à la chaîne à l'aide de la commande UNSUBSCRIBE en spécifiant la chaîne de laquelle il veut se désabonner.

```
UNSUBSCRIBE news.sports.golf
```

Example – Abonnements à plusieurs chaînes spécifiées

Pour vous abonner à plusieurs chaînes spécifiques, affichez la liste des chaînes avec la commande SUBSCRIBE. Dans l'exemple suivant, un client souscrit aux chaînes news.sports.golf, news.sports.soccer et news.sports.skiing.

```
SUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

Pour annuler un abonnement à une chaîne spécifique, utilisez la commande UNSUBSCRIBE et spécifiez la chaîne de laquelle vous voulez vous désabonner.

```
UNSUBSCRIBE news.sports.golf
```

Pour annuler les abonnements à plusieurs chaînes, utilisez la commande UNSUBSCRIBE et spécifiez les chaînes desquelles vous voulez vous désabonner.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer
```

Pour annuler tous les abonnements, utilisez `UNSUBSCRIBE` et spécifiez chaque canal. Ou utilisez `UNSUBSCRIBE` et ne spécifiez pas de canal.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

or

```
UNSUBSCRIBE
```

Example – Abonnements à l'aide la correspondance de modèles

Les clients peuvent souscrire à toutes les chaînes qui correspondent à un modèle à l'aide de la commande PSUBSCRIBE.

Dans l'exemple suivant, un client souscrit à toutes les chaînes de sport. Vous ne listez pas toutes les chaînes de sport individuellement, comme vous le faites à l'aide de `SUBSCRIBE`. Au lieu de cela, avec la commande `PSUBSCRIBE`, vous utilisez la correspondance des motifs.

```
PSUBSCRIBE news.sports.*
```

Example Annulation d'abonnements

Pour annuler des abonnements à ces chaînes, utilisez la commande `PUNSUBSCRIBE`.

```
PUNSUBSCRIBE news.sports.*
```

> **Important**
>
> - La chaîne du canal envoyée à une commande [P]SUBSCRIBE et celle envoyée à la commande [P]UNSUBSCRIBE doivent correspondre. Vous ne pouvez pas `PSUBSCRIBE` vers news.* et `PUNSUBSCRIBE` depuis news.sports.* ou `UNSUBSCRIBE` depuis news.sports.golf.
> - `PSUBSCRIBE`et ne `PUNSUBSCRIBE` sont pas disponibles pour ElastiCache Serverless.

Publication

Pour envoyer un message à tous les abonnés d'une chaîne, utilisez la commande `PUBLISH`, en spécifiant la chaîne et le message. L'exemple suivant publie le message, « C'est samedi et il fait beau. Je suis dirigé vers les liens. » vers la chaîne news.sports.golf.

```
PUBLISH news.sports.golf "It's Saturday and sunny. I'm headed to the links."
```

Un client ne peut pas publier sur une chaîne à laquelle il est abonné.

Pour plus d'informations, consultez Pub/Sub dans la documentation de Valkey.

Données de recommandation (hachages)

L'utilisation de l'INCR ou du DECR dans Valkey ou Redis OSS simplifie la compilation des recommandations. Chaque fois qu'un utilisateur « aime » un produit, vous incrémentez un compteur item:productID:like. Chaque fois qu'un utilisateur « n'aime pas » un produit, vous incrémentez un compteur item:productID:like. À l'aide de hachages, vous pouvez également tenir à jour une liste de tous ceux qui ont aimé ou non un produit.

Example – « J'aime » et « Je n'aime pas »

```
INCR item:38923:likes
HSET item:38923:ratings Susan 1
INCR item:38923:dislikes
HSET item:38923:ratings Tommy -1
```

ElastiCache Témoignages de clients

Pour découvrir comment des entreprises comme Airbnb, PBS, Esri et d'autres utilisent Amazon ElastiCache pour développer leurs activités grâce à une meilleure expérience client, consultez Comment les autres utilisent Amazon. ElastiCache

Vous pouvez également visionner les vidéos du didacticiel pour découvrir d'autres cas d'utilisation par des ElastiCache clients.

Commencer à utiliser Amazon ElastiCache

Utilisez le didacticiel pratique de cette section pour vous aider à démarrer et en savoir plus sur l'utilisation ElastiCache.

Rubriques

- Con ElastiCachefiguration
- Création d'un cache sans serveur Valkey
- Création d'un cache sans serveur Redis OSS
- Création d'un cache sans serveur Memcached
- Tutoriels : Débuter avec Python et ElastiCache
- Tutoriel : Configuration de Lambda pour un accès ElastiCache dans un VPC

Con ElastiCachefiguration

Pour utiliser le service ElastiCache Web, procédez comme suit.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Octroi d'un accès par programmation
- Configurez vos autorisations (nouveaux ElastiCache utilisateurs uniquement)
- Configurez EC2
- Octroi d'un accès au réseau à votre cache à partir d'un groupe de sécurité Amazon VPC
- Téléchargez et configurez l'accès en ligne de commande

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 l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

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

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

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte 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, consultez la section 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, octroyez 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.

Connexion en tant qu'utilisateur doté 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.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

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 au AWS CLI AWS SDKs, ou AWS APIs.	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 AWS SDKs, outils, et AWS APIs, voir Authentification IAM Identity Center dans le guide de référence AWS SDKs et Tools.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	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.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
		• Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations d'identif ication à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir 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 ElastiCache utilisateurs 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. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le 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 dans le 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.

Amazon ElastiCache 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. ElastiCache Pour créer un rôle lié à un service pour vous, utilisez la politique AWS-managed nommée. `AmazonElastiCacheFullAccess` 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 disposez des autorisations pour appeler `iam:createServiceLinkedRole` ou que vous avez créé le rôle lié à un service ElastiCache.

Pour plus d'informations, consultez les ressources suivantes :

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

Configurez EC2

Vous devrez configurer une EC2 instance à partir de laquelle vous vous connecterez à votre cache.

- Si vous ne possédez pas encore d' EC2 instance, découvrez comment configurer une EC2 instance ici : Amazon EC2 Getting Started Guide.
- Votre EC2 instance doit se trouver dans le même VPC et avoir les mêmes paramètres de groupe de sécurité que votre cache. Par défaut, Amazon ElastiCache crée un cache dans votre VPC par défaut et utilise le groupe de sécurité par défaut. Pour suivre ce didacticiel, assurez-vous que votre EC2 instance se trouve dans le VPC par défaut et possède le groupe de sécurité par défaut.

Octroi d'un accès au réseau à votre cache à partir d'un groupe de sécurité Amazon VPC

ElastiCache les clusters conçus par leurs soins utilisent le port 6379 pour les commandes Valkey et Redis OSS, tandis que les clusters ElastiCache sans serveur utilisent à la fois les ports 6379 et 6380.

Afin de connecter et d'exécuter correctement les commandes Valkey ou Redis OSS à partir de votre EC2 instance, votre groupe de sécurité doit autoriser l'accès à ces ports selon les besoins.

ElastiCache for Memcached utilise les ports 11211 et 11212 pour accepter les commandes Memcached. Afin de connecter et d'exécuter correctement les commandes Memcached à partir de votre EC2 instance, votre groupe de sécurité doit autoriser l'accès à ces ports.

1. Connectez-vous à la EC2 console Amazon AWS Command Line Interface et ouvrez-la.
2. Dans le volet de navigation, sous Réseau et sécurité, choisissez Groupes de sécurité.
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 à ElastiCache utiliser, ce groupe de sécurité sera nommé par défaut.
4. Cliquez sur l'onglet Entrant, puis :
   a. Choisissez Edit (Modifier).
   b. Choisissez Ajouter une règle.
   c. Dans la colonne Type, choisissez Règle TCP personnalisée.
   d. Si vous utilisez Valkey ou Redis OSS, tapez dans la zone Plage de ports. `6379`

      Si vous utilisez Memcached, tapez dans le champ Plage de ports. `11211`
   e. Dans le champ Source, choisissez Anywhere dont la plage de ports est comprise (0.0.0.0/0) afin que toute EC2 instance Amazon que vous lancez au sein de votre Amazon VPC puisse se connecter à votre cache.
   f. Si vous utilisez le mode ElastiCache sans serveur, ajoutez une autre règle en choisissant Ajouter une règle.
   g. Dans la colonne Type, choisissez Règle TCP personnalisée.
   h. Si vous utilisez ElastiCache Redis OSS, tapez `6380` dans la zone Plage de ports.

      Si vous utilisez ElastiCache Memcached, tapez dans le champ Plage de ports. `11212`
   i. Dans le champ Source, choisissez Anywhere dont la plage de ports est comprise (0.0.0.0/0) afin que toute EC2 instance Amazon que vous lancez au sein de votre Amazon VPC puisse se connecter à votre cache.
   j. Choisissez Enregistrer.

Téléchargez et configurez l'accès en ligne de commande

Téléchargez et installez l'utilitaire valkey-cli.

Si vous utilisez ElastiCache for Valkey, l'utilitaire valkey-cli pourrait vous être utile. Si vous utilisez ElastiCache Redis OSS avec redis-cli, pensez à passer à valkey-cli car cela fonctionne également pour Redis OSS.

1. Connectez-vous à votre EC2 instance Amazon à l'aide de l'utilitaire de connexion de votre choix. Pour savoir comment vous connecter à une EC2 instance Amazon, consultez le guide de EC2 démarrage Amazon.
2. Téléchargez et installez l'utilitaire valkey-cli en exécutant la commande appropriée à votre configuration.

   Amazon Linux 2

   ```
   sudo amazon-linux-extras install epel -y
   sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
   wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
   tar xvzf valkey-8.0.0.tar.gz
   cd valkey-8.0.0
   make BUILD_TLS=yes
   ```

> **Note**
>
> - Lorsque vous installez le package redis6, l'utilitaire redis6-cli est installé avec la prise en charge du chiffrement par défaut.
> - Il est important de disposer d'un support de compilation pour TLS lors de l'installation de valkey-cli ou de redis-cli. ElastiCache Le mode Serverless n'est accessible que lorsque le protocole TLS est activé.
> - Si vous vous connectez à un cluster qui n'est pas chiffré, vous n'avez pas besoin de l'option `Build_TLS=yes`.

Création d'un cache sans serveur Valkey

Au cours de cette étape, vous allez créer un nouveau cache sur Amazon ElastiCache.

AWS Management Console

Pour créer un nouveau cache à l'aide de la ElastiCache console :

1. Connectez-vous au AWS Management Console et ouvrez le https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Valkey caches.
3. Sur le côté droit de la console, choisissez Create Valkey cache
4. Dans Paramètres de cache, entrez un Nom. Vous pouvez éventuellement entrer une description pour le cache.
5. Laissez les paramètres par défaut sélectionnés.
6. Cliquez sur Créer pour créer le cache.
7. Une fois que le cache est à l'état « ACTIVE », vous pouvez commencer à écrire et à lire des données sur le cache.

AWS CLI

L' AWS CLI exemple suivant crée un nouveau cache à l'aide de create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine valkey
```

Windows

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name CacheName ^
    --engine valkey
```

Notez que la valeur du champ Statut est définie sur `CREATING`.

Pour vérifier que la création du cache ElastiCache est terminée, utilisez la `describe-serverless-caches` commande.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Une fois le nouveau cache créé, passez à l'Lire et écrire des données dans le cache.

Lire et écrire des données dans le cache

Cette section part du principe que vous avez créé une EC2 instance Amazon et que vous pouvez vous y connecter. Pour savoir comment procéder, consultez le guide de EC2 démarrage Amazon.

Cette section suppose également que vous avez configuré les paramètres d'accès au VPC et de groupe de sécurité pour l' EC2 instance à partir de laquelle vous vous connectez à votre cache, et que vous avez configuré valkey-cli sur votre instance. EC2 Pour plus d'informations sur cette étape, consultez Con ElastiCachefiguration.

Outre les étapes ci-dessous, si votre application est volumineuse ou globale, vous pouvez améliorer considérablement les performances de lecture en créant et en lisant des répliques. Pour plus d'informations sur cette étape plus avancée, voirBonnes pratiques d'utilisation des répliques de lecture.

Recherche du point de terminaison de votre cache

AWS Management Console

Pour trouver le point de terminaison de votre cache à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Valkey caches.
3. À droite de la console, cliquez sur le nom du cache que vous venez de créer.
4. Dans Détails du cache, localisez et copiez le point de terminaison du cache.

AWS CLI

L' AWS CLI exemple suivant montre comment rechercher le point de terminaison de votre nouveau cache à l'aide de la describe-serverless-caches commande. Une fois que vous avez exécuté la commande, recherchez le champ « Endpoint ».

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Connectez-vous à votre Valkey Cache (Linux)

Maintenant que vous avez le point de terminaison dont vous avez besoin, vous pouvez vous connecter à votre EC2 instance et vous connecter au cache. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La commande suivante permet de se connecter à un cache (remarque : remplacez cache-endpoint par le point de terminaison que vous avez récupéré à l'étape précédente).

```
src/valkey-cli -h cache-endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Connectez-vous à votre Valkey Cache (Windows)

Maintenant que vous avez le point de terminaison dont vous avez besoin, vous pouvez vous connecter à votre EC2 instance et vous connecter au cache. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La commande suivante permet de se connecter à un cache. Ouvrez l'invite de commande, accédez au répertoire Valkey ou Redis OSS et exécutez la commande (remarque : remplacez Cache_Endpoint par le point de terminaison que vous avez récupéré à l'étape précédente).

```
c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Vous pouvez maintenant passer à l'(Facultatif) Nettoyer.

(Facultatif) Nettoyer

Si vous n'avez plus besoin du ElastiCache cache Amazon que vous avez créé, vous pouvez le supprimer. Cette étape permet de vous assurer de ne pas être facturé pour des ressources que vous n'utilisez pas. Vous pouvez utiliser la ElastiCache console AWS CLI, l'API ou l' ElastiCache API pour supprimer votre cache.

AWS Management Console

Pour supprimer votre cache à l'aide de la console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Valkey Caches.
3. Cliquez sur le bouton d'option en regard du cache que vous voulez supprimer.
4. Sélectionnez Actions en haut à droite, puis Supprimer.
5. Vous pouvez éventuellement choisir de prendre un instantané final avant de supprimer votre cache.
6. Dans l'écran de confirmation Supprimer, entrez à nouveau le nom du cache et choisissez Supprimer pour supprimer le cluster ou Annuler pour le conserver.

Dès que votre cache passe au statut DELETING, vous cessez d'être facturé pour celui-ci.

AWS CLI

L' AWS CLI exemple suivant supprime un cache à l'aide de la delete-serverless-cache commande.

Linux

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^
  --serverless-cache-name CacheName
```

Notez que la valeur du champ Statut est définie sur DELETING.

Vous pouvez maintenant passer à l'Étapes suivantes.

Étapes suivantes

Pour plus d'informations, ElastiCache consultez les pages suivantes :

- Travailler avec ElastiCache
- Dimensionnement ElastiCache
- Journalisation et surveillance sur Amazon ElastiCache
- ElastiCache meilleures pratiques et stratégies de mise en cache
- Instantané et restauration
- Surveillance des événements par Amazon SNS ElastiCache

Création d'un cache sans serveur Redis OSS

Au cours de cette étape, vous allez créer un nouveau cache sur Amazon ElastiCache.

AWS Management Console

Pour créer un nouveau cache à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez les caches Redis OSS.
3. Sur le côté droit de la console, choisissez Create Redis OSS cache
4. Dans Paramètres de cache, entrez un Nom. Vous pouvez éventuellement entrer une description pour le cache.
5. Laissez les paramètres par défaut sélectionnés.
6. Cliquez sur Créer pour créer le cache.
7. Une fois que le cache est à l'état « ACTIVE », vous pouvez commencer à écrire et à lire des données sur le cache.

AWS CLI

L' AWS CLI exemple suivant crée un nouveau cache à l'aide de create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine redis
```

Windows

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name CacheName ^
    --engine redis
```

Notez que la valeur du champ Statut est définie sur `CREATING`.

Pour vérifier que la création du cache ElastiCache est terminée, utilisez la `describe-serverless-caches` commande.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Une fois le nouveau cache créé, passez à l'Lire et écrire des données dans le cache.

Lire et écrire des données dans le cache

Cette section part du principe que vous avez créé une EC2 instance Amazon et que vous pouvez vous y connecter. Pour obtenir des instructions sur la procédure à suivre, consultez le guide de EC2 démarrage Amazon.

Cette section suppose également que vous avez configuré les paramètres d'accès au VPC et de groupe de sécurité pour l' EC2 instance à partir de laquelle vous vous connectez à votre cache, et que vous avez configuré valkey-cli sur votre instance. EC2 Pour plus d'informations sur cette étape, consultez Con ElastiCachefiguration.

Recherche du point de terminaison de votre cache

AWS Management Console

Pour trouver le point de terminaison de votre cache à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Valkey caches Redis OSS caches.
3. À droite de la console, cliquez sur le nom du cache que vous venez de créer.
4. Dans Détails du cache, localisez et copiez le point de terminaison du cache.

AWS CLI

L' AWS CLI exemple suivant montre comment rechercher le point de terminaison de votre nouveau cache à l'aide de la describe-serverless-caches commande. Une fois que vous avez exécuté la commande, recherchez le champ « Endpoint ».

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Connectez-vous à votre cache Valkey ou Redis OSS (Linux)

Maintenant que vous avez le point de terminaison dont vous avez besoin, vous pouvez vous connecter à votre EC2 instance et vous connecter au cache. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La commande suivante permet de se connecter à un cache (remarque : remplacez cache-endpoint par le point de terminaison que vous avez récupéré à l'étape précédente).

```
src/valkey-cli -h cache-endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Connectez-vous à votre cache Valkey ou Redis OSS (Windows)

Maintenant que vous avez le point de terminaison dont vous avez besoin, vous pouvez vous connecter à votre EC2 instance et vous connecter au cache. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La commande suivante permet de se connecter à un cache. Ouvrez l'invite de commande, accédez au répertoire Valkey et exécutez la commande (remarque : remplacez Cache_Endpoint par le point de terminaison que vous avez récupéré à l'étape précédente).

```
c:\Redis>valkey-cli -h Redis_Cluster_Endpoint --tls -p 6379
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
```

Vous pouvez maintenant passer à l'(Facultatif) Nettoyer.

(Facultatif) Nettoyer

Si vous n'avez plus besoin du ElastiCache cache Amazon que vous avez créé, vous pouvez le supprimer. Cette étape permet de vous assurer de ne pas être facturé pour des ressources que vous n'utilisez pas. Vous pouvez utiliser la ElastiCache console AWS CLI, l'API ou l' ElastiCache API pour supprimer votre cache.

AWS Management Console

Pour supprimer votre cache à l'aide de la console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation sur le côté gauche de la console, choisissez Valkey ou Redis OSS Caches.
3. Cliquez sur le bouton d'option en regard du cache que vous voulez supprimer.
4. Sélectionnez Actions en haut à droite, puis Supprimer.
5. Vous pouvez éventuellement choisir de prendre un instantané final avant de supprimer votre cache.
6. Dans l'écran de confirmation Supprimer, entrez à nouveau le nom du cache et choisissez Supprimer pour supprimer le cluster ou Annuler pour le conserver.

Dès que votre cache passe au statut DELETING, vous cessez d'être facturé pour celui-ci.

AWS CLI

L' AWS CLI exemple suivant supprime un cache à l'aide de la delete-serverless-cache commande.

Linux

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^
  --serverless-cache-name CacheName
```

Notez que la valeur du champ Statut est définie sur DELETING.

Vous pouvez maintenant passer à l'Étapes suivantes.

Étapes suivantes

Pour plus d'informations, ElastiCache consultez les pages suivantes :

- Travailler avec ElastiCache
- Dimensionnement ElastiCache
- Journalisation et surveillance sur Amazon ElastiCache
- ElastiCache meilleures pratiques et stratégies de mise en cache
- Instantané et restauration
- Surveillance des événements par Amazon SNS ElastiCache

Création d'un cache sans serveur Memcached

AWS Management Console

Pour créer un nouveau cache sans serveur Memcached à l'aide de la console : ElastiCache

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

2. Dans le volet de navigation à gauche de la console, choisissez Caches Memcached.
3. Dans la partie droite de la console, choisissez Créer un cache Memcached.
4. Dans Paramètres de cache, entrez un Nom. Vous pouvez éventuellement entrer une description pour le cache.
5. Laissez les paramètres par défaut sélectionnés.
6. Cliquez sur Créer pour créer le cache.
7. Une fois que le cache est à l'état « ACTIVE », vous pouvez commencer à écrire et à lire des données sur le cache.

Pour créer un nouveau cache à l'aide du AWS CLI

L' AWS CLI exemple suivant crée un nouveau cache à l'aide de create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \
  --serverless-cache-name CacheName \
  --engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^
  --serverless-cache-name CacheName ^
  --engine memcached
```

Notez que la valeur du champ Statut est définie sur `CREATING`.

Pour vérifier que la création du cache ElastiCache est terminée, utilisez la `describe-serverless-caches` commande.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Une fois le nouveau cache créé, passez à l'Lire et écrire des données dans le cache.

Lire et écrire des données dans le cache

Cette section part du principe que vous avez créé une EC2 instance Amazon et que vous pouvez vous y connecter. Pour obtenir des instructions sur la procédure à suivre, consultez le guide de EC2 démarrage Amazon.

Par défaut, ElastiCache crée un cache dans votre VPC par défaut. Assurez-vous que votre EC2 instance est également créée dans le VPC par défaut, afin qu'elle puisse se connecter au cache.

Recherche du point de terminaison de votre cache

AWS Management Console

Pour trouver le point de terminaison de votre cache à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation à gauche de la console, choisissez Caches Memcached.
3. À droite de la console, cliquez sur le nom du cache que vous venez de créer.
4. Dans Détails du cache, localisez et copiez le point de terminaison du cache.

AWS CLI

L' AWS CLI exemple suivant montre comment rechercher le point de terminaison de votre nouveau cache à l'aide de la describe-serverless-caches commande. Une fois que vous avez exécuté la commande, recherchez le champ « Endpoint ».

Linux

```
aws elasticache describe-serverless-caches \
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^
  --serverless-cache-name CacheName
```

Connexion à l'aide d'OpenSSL

Pour en savoir plus sur la connexion à l'aide d'OpenSSL, consultez ElastiCache chiffrement en transit (TLS).

Connexion à l'aide du client Java Memcached

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.FailureMode;
import net.spy.memcached.MemcachedClient;

public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
 ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
 TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        // Set Failure Mode to Retry
        connectionFactoryBuilder.setFailureMode(FailureMode.Retry);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
 AddrUtil.getAddresses("mycluster-fnjyzo.serverless.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Connexion à l'aide du client PHP Memcached

```
<?php
$cluster_endpoint = "mycluster.serverless.use1.cache.amazonaws.com";
$server_port = 11211;
```

```
/* Initialize a persistent Memcached client in TLS mode */
$tls_client = new Memcached('persistent-id');
$tls_client->addServer($cluster_endpoint, $server_port);
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.serverless.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
$tls_client->createAndSetTLSContext((array)$tls_config);

 /* store the data for 60 seconds in the cluster */
$tls_client->set('key', 'value', 60);
?>
```

Connexion à l'aide du client Python Memcached (Pymemcache)

Voir https://pymemcache.readthedocs. io/en/latest/getting_started.html

```
import ssl
from pymemcache.client.base import Client

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
target_port = 11211
memcached_client = Client(("{cluster_endpoint}", target_port), tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

Connectez-vous à l'aide du NodeJS/TS client Memcached (Memcache Electrode-IO)

Voir https://github.com/electrode-io/memcache et memcache-client https://www.npmjs.com/package/

Procédez à l'installation via `npm i memcache-client`

Dans l'application, créez un client TLS Memcached comme suit :

```
var memcache = require("memcache-client");
const client = new memcache.MemcacheClient({server: "{cluster_endpoint}:11211", tls:
 {}});
client.set("key", "value");
```

Connexion à l'aide du client Rust Memcached (rust-memcache)

Voir https://crates. io/crates/memcacheet https://github.com/aisk/rust-memcache.

```
// create connection with to memcached server node:
let client = memcache::connect("memcache+tls://<cluster_endpoint>:11211?
verify_mode=none").unwrap();

// set a string value
client.set("foo", "bar", 0).unwrap();
```

Connexion à l'aide du client Go Memcached (Gomemcache)

Voir https://github.com/bradfitz/gomemcache

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

Connexion à l'aide du client Ruby Memcached (Dalli)

Voir https://github.com/petergoldstein/Dalli

```
require 'dalli'
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_context.ssl_version = :SSLv23
ssl_context.verify_hostname = true
ssl_context.verify_mode = OpenSSL::SSL::VERIFY_PEER
client = Dalli::Client.new("<cluster_endpoint>:11211", :ssl_context => ssl_context);
client.get("abc")
```

Connect à l'aide du client .NET Memcached () EnyimMemcachedCore

Consultez https://github.com/cnblogs/EnyimMemcachedCore.

```
"MemcachedClient": {
"Servers": [
{
"Address": "{cluster_endpoint}",
"Port": 11211
}
],
"UseSslStream":  true
}
```

Vous pouvez maintenant passer à l'(Facultatif) Nettoyer.

(Facultatif) Nettoyer

À l'aide du AWS Management Console

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

Pour supprimer un cache

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le tableau de bord de la ElastiCache console, choisissez le moteur qui s'exécute sur le cache que vous souhaitez supprimer. La liste de tous les caches exécutant le moteur choisi s'affiche.
3. Pour choisir le cache à supprimer, choisissez son nom dans la liste des caches.

   > **Important**
   >
   > Vous ne pouvez supprimer qu'un seul cache à la fois depuis la ElastiCache console. Le choix de plusieurs caches désactive l'opération de suppression.

4. Pour Actions, choisissez Supprimer.
5. Dans l'écran de confirmation Supprimer le cache, choisissez Supprimer pour supprimer le cache ou Annuler pour le conserver.
6. Si vous avez choisi Supprimer, le statut du cache passe à deleting.

Dès que votre cache passe au statut DELETING, vous cessez d'être facturé pour celui-ci.

À l'aide du AWS CLI

Le code suivant supprime le cache my-cache.

```
aws elasticache delete-serverless-cache --serverless-cache-name my-cache
```

L'action delete-serverless-cache CLI ne supprime qu'un seul cache sans serveur. Pour supprimer plusieurs caches, appelez delete-serverless-cache chaque cache sans serveur que vous souhaitez

supprimer. Vous n'avez pas besoin d'attendre que la suppression d'un cache sans serveur soit terminée avant d'en supprimer un autre.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-serverless-cache \
  --serverless-cache-name my-cache
```

Pour Windows:

```
aws elasticache delete-serverless-cache ^
  --serverless-cache-name my-cache
```

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée delete-serverless-cache.

Vous pouvez maintenant passer à l'Étapes suivantes.

Étapes suivantes

Pour plus d'informations ElastiCache sur :

- Travailler avec ElastiCache
- Dimensionnement ElastiCache
- Quotas pour ElastiCache
- ElastiCache meilleures pratiques et stratégies de mise en cache
- Affichage des ElastiCache événements

Tutoriels : Débuter avec Python et ElastiCache

Cette section contient des didacticiels pratiques pour vous aider à en savoir plus sur ElastiCache Valkey et Redis OSS. Nous vous invitons à utiliser l'un de nos didacticiels spécifiques au langage.

> **Note**
>
> AWS SDKs sont disponibles dans une grande variété de langues. Pour obtenir la liste complète, consultez Outils d'Amazon Web Services.

Rubriques

- Python et ElastiCache

Python et ElastiCache

Dans ce didacticiel, vous allez utiliser le AWS SDK pour Python (Boto3) pour écrire des programmes simples permettant d'effectuer les opérations suivantes : ElastiCache

- Créer ElastiCache pour les clusters Redis OSS (mode cluster activé et mode cluster désactivé)
- Vérifiez si des utilisateurs ou des groupes d'utilisateurs existent, sinon créez-les. (Cette fonctionnalité est disponible avec Valkey 7.2 et versions ultérieures, et avec Redis OSS 6.0 à 7.1.)
- Connect à ElastiCache
- Effectuez des opérations telles que le réglage et l'obtention de chaînes, la lecture et l'écriture sur des streams, ainsi que la publication et l'abonnement à Pub/Sub une chaîne.

Au cours de ce didacticiel, vous pouvez consulter la documentation du AWS SDK pour Python (Boto). La section suivante est spécifique à ElastiCache : client de ElastiCache bas niveau

Prérequis du didacticiel

- Configurez une clé AWS d'accès pour utiliser le AWS SDKs. Pour de plus amples informations, veuillez consulter Con ElastiCachefiguration.
- Installer Python 3.0 ou version ultérieure. Pour plus d'informations, rendez-vous sur https://www.python.org/downloads. Pour plus d'informations, consultez la page Quickstart dans la documentation Boto 3.

Rubriques

- Tutoriel : Création de ElastiCache clusters et d'utilisateurs
- Tutoriel : Connexion à ElastiCache
- Exemples d'utilisation

Tutoriel : Création de ElastiCache clusters et d'utilisateurs

Les exemples suivants utilisent le SDK boto3 ElastiCache pour les opérations de gestion Redis OSS (création de clusters ou d'utilisateurs) et redis-py-cluster redis-py/ pour le traitement des données.

Rubriques

- Créer un cluster en mode cluster désactivé
- Créer un cluster en mode cluster désactivé avec TLS et RBAC
- Créer un cluster Redis en mode activé
- Créer un cluster en mode cluster activé avec TLS et RBAC
- Vérifiez s'ils users/usergroup existent, sinon créez-les

Créer un cluster en mode cluster désactivé

Copiez le programme suivant et collez-le dans un fichier nommé CreateClusterModeDisabledCluster.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_disabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheCluste
 cache cluster',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
 node) and maximun 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'
```

```
    response = client.create_replication_group(
        AutomaticFailoverEnabled=True,
        CacheNodeType=CacheNodeType,
        Engine='valkey',
        EngineVersion=EngineVersion,
        NumCacheClusters=NumCacheClusters,
        ReplicationGroupDescription=ReplicationGroupDescription,
        ReplicationGroupId=ReplicationGroupId,
        SnapshotRetentionLimit=30,
    )
    return response


if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
 nodes, Valkey 8.0, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104053'
        )

    logging.info(elasticacheResponse)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python CreateClusterModeDisabledCluster.py
```

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

Créer un cluster en mode cluster désactivé avec TLS et RBAC

Pour garantir la sécurité, vous pouvez utiliser TLS (Transport Layer Security) et RBAC (Role Based Access Control) lors de la création d'un cluster en mode cluster désactivé. Contrairement à Valkey ou Redis OSS AUTH, où tous les clients authentifiés ont un accès complet au groupe de réplication si leur jeton est authentifié, le RBAC vous permet de contrôler l'accès au cluster par le biais de groupes d'utilisateurs. Ces groupes d'utilisateurs sont conçus comme un moyen d'organiser l'accès aux groupes de réplication. Pour de plus amples informations, veuillez consulter Contrôle d'accès basé sur les rôles (RBAC).

Copiez le programme suivant et collez-le dans un fichier nommé ClusterModeDisabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumCacheC
 cache cluster',ReplicationGroupId=None, UserGroupIds=None,
 SecurityGroupIds=None,CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
 node) and maximun 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Mandatory name for the cluster.
    :param UserGroupIds: The ID of the user group to be assigned to the cluster.
    :param SecurityGroupIds: List of security groups to be assigned. If not defined,
 default will be used
    :param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
 defined, default will be used.
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return {'Error': 'ReplicationGroupId parameter is required'}
    elif not isinstance(UserGroupIds,(list)):
        return {'Error': 'UserGroupIds parameter is required and must be a list'}

    params={'AutomaticFailoverEnabled': True,
            'CacheNodeType': CacheNodeType,
            'Engine': 'valkey',
            'EngineVersion': EngineVersion,
```

```
            'NumCacheClusters': NumCacheClusters,
            'ReplicationGroupDescription': ReplicationGroupDescription,
            'ReplicationGroupId': ReplicationGroupId,
            'SnapshotRetentionLimit': 30,
            'TransitEncryptionEnabled': True,
            'UserGroupIds':UserGroupIds
        }

    # defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
    if isinstance(SecurityGroupIds,(list)):
        params.update({'SecurityGroupIds':SecurityGroupIds})
    if CacheSubnetGroupName:
        params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

    response = client.create_replication_group(**params)
    return response

if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
 nodes, Valkey 8.0, one primary and two replicas.
    # Assigns the existent user group "mygroup" for RBAC authentication

    response=create_cluster_mode_disabled_rbac(
        CacheNodeType='cache.m6g.large',
        EngineVersion='8.0',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104',
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'
    )

    logging.info(response)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ClusterModeDisabledWithRBAC.py
```

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

Créer un cluster Redis en mode activé

Copiez le programme suivant et collez-le dans un fichier nommé ClusterModeEnabled.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_enabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumNodeGroups=1
 ReplicationGroupDescription='Sample cache with cluster mode
 enabled',ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode enabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximun 90.
    If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
 replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'

    response = client.create_replication_group(
        AutomaticFailoverEnabled=True,
        CacheNodeType=CacheNodeType,
        Engine='valkey',
        EngineVersion=EngineVersion,
        ReplicationGroupDescription=ReplicationGroupDescription,
```

```
        ReplicationGroupId=ReplicationGroupId,
    #   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 node (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=NumNodeGroups,
        ReplicasPerNodeGroup=ReplicasPerNodeGroup,
        CacheParameterGroupName='default.valkey7.2.cluster.on'
    )

    return response


# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
    ReplicationGroupId='valkey20210',
#   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=2,
    ReplicasPerNodeGroup=1,
)

logging.info(response)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ClusterModeEnabled.py
```

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

Créer un cluster en mode cluster activé avec TLS et RBAC

Pour garantir la sécurité, vous pouvez utiliser TLS (Transport Layer Security) et RBAC (Role Based Access Control) lors de la création d'un cluster en mode cluster activé. Contrairement à Valkey ou Redis OSS AUTH, où tous les clients authentifiés ont un accès complet au groupe de réplication si leur jeton est authentifié, le RBAC vous permet de contrôler l'accès au cluster par le biais de groupes d'utilisateurs. Ces groupes d'utilisateurs sont conçus comme un moyen d'organiser l'accès aux groupes de réplication. Pour de plus amples informations, veuillez consulter Contrôle d'accès basé sur les rôles (RBAC).

Copiez le programme suivant et collez-le dans un fichier nommé ClusterModeEnabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
 create_cluster_mode_enabled(CacheNodeType='cache.t3.small',EngineVersion='6.0',NumNodeGroups=1
 ReplicationGroupDescription='Sample cache with cluster
 mode enabled',ReplicationGroupId=None,UserGroupIds=None,
 SecurityGroupIds=None,CacheSubnetGroupName=None,CacheParameterGroupName='default.valkey7.2.clu
    """Creates an ElastiCache Cluster with cluster mode enabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
 cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
 used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximun 90.
    If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
 replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
    :param ReplicationGroupId: Name for the cluster.
    :param CacheParameterGroupName: Parameter group to be used. Must be compatible with
 the engine version and cluster mode enabled.
    :return: dictionary with the API results

    """
    if not ReplicationGroupId:
        return 'ReplicationGroupId parameter is required'
    elif not isinstance(UserGroupIds,(list)):
        return {'Error': 'UserGroupIds parameter is required and must be a list'}

    params={'AutomaticFailoverEnabled': True,
            'CacheNodeType': CacheNodeType,
            'Engine': 'valkey',
            'EngineVersion': EngineVersion,
            'ReplicationGroupDescription': ReplicationGroupDescription,
            'ReplicationGroupId': ReplicationGroupId,
```

```
            'SnapshotRetentionLimit': 30,
            'TransitEncryptionEnabled': True,
            'UserGroupIds':UserGroupIds,
            'NumNodeGroups': NumNodeGroups,
            'ReplicasPerNodeGroup': ReplicasPerNodeGroup,
            'CacheParameterGroupName': CacheParameterGroupName
        }

    # defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
    if isinstance(SecurityGroupIds,(list)):
        params.update({'SecurityGroupIds':SecurityGroupIds})
    if CacheSubnetGroupName:
        params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

    response = client.create_replication_group(**params)
    return response

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
        CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
        ReplicationGroupId='valkey2021',
    #   Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
 (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=2,
        ReplicasPerNodeGroup=1,
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'

    )

    logging.info(response)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ClusterModeEnabledWithRBAC.py
```

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

Vérifiez s'ils users/usergroup existent, sinon créez-les

Avec RBAC, vous créez des utilisateurs et leur attribuez des autorisations spécifiques à l'aide d'une chaîne d'accès. Vous assignez les utilisateurs à des groupes d'utilisateurs correspondant à un rôle spécifique (administrateurs, ressources humaines) qui sont ensuite déployés dans un ou plusieurs groupes ElastiCache de réplication Redis OSS. Vous pouvez ainsi établir des limites de sécurité entre les clients utilisant le ou les mêmes groupes de réplication Valkey ou Redis OSS et empêcher les clients d'accéder aux données des autres. Pour de plus amples informations, veuillez consulter Contrôle d'accès basé sur les rôles (RBAC).

Copiez le programme suivant et collez-le dans un fichier nommé UserAndUserGroups.py. Mettez à jour le mécanisme de fourniture des informations d'identification. Dans cet exemple, les informations d'identification sont affichées comme remplaçables et un élément non déclaré leur est affecté. Évitez de coder en dur les informations d'identification.

Cet exemple utilise une chaîne d'accès contenant les autorisations de l'utilisateur. Pour plus d'informations sur les chaînes d'accès, voirDéfinition des autorisations à l'aide d'une chaîne d'accès.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def check_user_exists(UserId):
    """Checks if UserId exists

    Returns True if UserId exists, otherwise False
    :param UserId: ElastiCache User ID
    :return: True|False
    """
    try:
        response = client.describe_users(
            UserId=UserId,
        )
        if response['Users'][0]['UserId'].lower() == UserId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserNotFound':
            logging.info(e.response['Error'])
```

```
            return False
        else:
            raise

def check_group_exists(UserGroupId):
    """Checks if UserGroupID exists

    Returns True if Group ID exists, otherwise False
    :param UserGroupId: ElastiCache User ID
    :return: True|False
    """

    try:
        response = client.describe_user_groups(
            UserGroupId=UserGroupId
        )
        if response['UserGroups'][0]['UserGroupId'].lower() == UserGroupId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserGroupNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def create_user(UserId=None,UserName=None,Password=None,AccessString=None):
    """Creates a new user

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param UserName: ElastiCache user name. ElastiCache allows multiple users with the
 same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
    :param AccessString: Access string with the permissions for the user.
    :return: user ARN
    """
    try:
        response = client.create_user(
            UserId=UserId,
            UserName=UserName,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
```

```
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
        return e.response['Error']

def create_group(UserGroupId=None, UserIds=None):
    """Creates a new group.
    A default user is required (mandatory) and should be specified in the UserIds list

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
 user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
            Engine='Redis',
            UserIds=UserIds
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])


if __name__ == '__main__':

    groupName='mygroup2'
    userName = 'myuser2'
    userId=groupName+'-'+userName

    # Creates a new user if the user ID does not exist.
    for tmpUserId,tmpUserName in [ (userId,userName), (groupName+'-
default','default')]:
        if not check_user_exists(tmpUserId):
            response=create_user(UserId=tmpUserId,
 UserName=EXAMPLE,Password=EXAMPLE,AccessString='on ~* +@all')
            logging.info(response)
        # assigns the new user ID to the user group
    if not check_group_exists(groupName):
        UserIds = [ userId , groupName+'-default']
        response=create_group(UserGroupId=groupName,UserIds=UserIds)
```

```
        logging.info(response)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python UserAndUserGroups.py
```

Tutoriel : Connexion à ElastiCache

Les exemples suivants utilisent le client Valkey ou Redis OSS pour se connecter à. ElastiCache

Rubriques

- Connexion à un cluster en mode cluster désactivé
- Connexion à un cluster en mode cluster activé

Connexion à un cluster en mode cluster désactivé

Copiez le programme suivant et collez-le dans un fichier nommé ConnectClusterModeDisabled.py. Mettez à jour le mécanisme de fourniture des informations d'identification. Dans cet exemple, les informations d'identification sont affichées comme remplaçables et un élément non déclaré leur est affecté. Évitez de coder en dur les informations d'identification.

```
from redis import Redis
import logging

logging.basicConfig(level=logging.INFO)
redis = Redis(host='primary.xxx.yyyyyy.zzz1.cache.amazonaws.com', port=6379,
 decode_responses=True, ssl=True, username=example, password=EXAMPLE)

if redis.ping():
    logging.info("Connected to Redis")
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ConnectClusterModeDisabled.py
```

Connexion à un cluster en mode cluster activé

Copiez le programme suivant et collez-le dans un fichier nommé ConnectClusterModeEnabled.py.

```
from rediscluster import RedisCluster
import logging

logging.basicConfig(level=logging.INFO)
redis = RedisCluster(startup_nodes=[{"host":
 "xxx.yyy.clustercfg.zzz1.cache.amazonaws.com","port": "6379"}],
 decode_responses=True,skip_full_coverage_check=True)

if redis.ping():
    logging.info("Connected to Redis")
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ConnectClusterModeEnabled.py
```

Exemples d'utilisation

Les exemples suivants utilisent le SDK boto3 ElastiCache pour fonctionner avec ElastiCache Redis OSS.

Rubriques

- Définir et obtenir des chaînes
- Définir et obtenir un hachage avec plusieurs éléments
- Publier (écrire) et s'abonner (lire) depuis une Pub/Sub chaîne
- Écrire et lire à partir d'un flux

Définir et obtenir des chaînes

Copiez le programme suivant et collez-le dans un fichier nommé SetAndGetStrings.py.

```
import time
import logging
logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
currTime=time.ctime(time.time())

# Set the key 'mykey' with the current date and time as value.
# The Key will expire and removed from cache in 60 seconds.
redis.set(keyName, currTime, ex=60)
```

```
# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieve the key value and current TTL
keyValue=redis.get(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
 keyValue, keyTTL))
```

Pour exécuter le programme, saisissez la commande suivante.

```
python SetAndGetStrings.py
```

Définir et obtenir un hachage avec plusieurs éléments

Copiez le programme suivant et collez-le dans un fichier nommé SetAndGetHash.py.

```
import logging
import time

logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
keyValues={'datetime': time.ctime(time.time()), 'epochtime': time.time()}

# Set the hash 'mykey' with the current date and time in human readable format
 (datetime field) and epoch number (epochtime field).
redis.hset(keyName, mapping=keyValues)

# Set the key to expire and removed from cache in 60 seconds.
redis.expire(keyName, 60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieves all the fields and current TTL
keyValues=redis.hgetall(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
 keyValues, keyTTL))
```

Pour exécuter le programme, saisissez la commande suivante.

```
python SetAndGetHash.py
```

Publier (écrire) et s'abonner (lire) depuis une Pub/Sub chaîne

Copiez le programme suivant et collez-le dans un fichier nommé PubAndSub.py.

```
import logging
import time

def handlerFunction(message):
    """Prints message got from PubSub channel to the log output

    Return None
    :param message: message to log
    """
    logging.info(message)

logging.basicConfig(level=logging.INFO)
redis = Redis(host="redis202104053.tihewd.ng.0001.use1.cache.amazonaws.com", port=6379,
 decode_responses=True)


# Creates the subscriber connection on "mychannel"
subscriber = redis.pubsub()
subscriber.subscribe(**{'mychannel': handlerFunction})

# Creates a new thread to watch for messages while the main process continues with its
 routines
thread = subscriber.run_in_thread(sleep_time=0.01)

# Creates publisher connection on "mychannel"
redis.publish('mychannel', 'My message')

# Publishes several messages. Subscriber thread will read and print on log.
while True:
    redis.publish('mychannel',time.ctime(time.time()))
    time.sleep(1)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python PubAndSub.py
```

Écrire et lire à partir d'un flux

Copiez le programme suivant et collez-le dans un fichier nommé ReadWriteStream.py.

```
from redis import Redis
import redis.exceptions as exceptions
import logging
import time
import threading

logging.basicConfig(level=logging.INFO)

def writeMessage(streamName):
    """Starts a loop writting the current time and thread name to 'streamName'

    :param streamName: Stream (key) name to write messages.
    """
    fieldsDict={'writerId':threading.currentThread().getName(),'myvalue':None}
    while True:
        fieldsDict['myvalue'] = time.ctime(time.time())
        redis.xadd(streamName,fieldsDict)
        time.sleep(1)

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
 used to coordinate data distribution.
    Once a thread acknowleges the message, it won't be provided again. If message
 wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group were multiple threads will read.
    :param streamName: Stream (key) name where messages will be read.
    """

    readerID=threading.currentThread().getName()
    while True:
        try:
            # Check if the stream has any message
            if redis.xlen(streamName)>0:
                # Check if if the messages are new (not acknowledged) or not (already
 processed)
                streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
                if len(streamData) > 0:
```

```
                    msgId,message = streamData[0][1][0]
                    logging.info("{}: Got {} from ID
 {}".format(readerID,message,msgId))
                    #Do some processing here. If the message has been processed
 sucessfuly, acknowledge it and (optional) delete the message.
                    redis.xack(streamName,groupName,msgId)
                    logging.info("Stream message ID {} read and processed successfuly
 by {}".format(msgId,readerID))
                    redis.xdel(streamName,msgId)
            else:
                pass
        except:
            raise

        time.sleep(0.5)

# Creates the stream 'mystream' and consumer group 'myworkergroup' where multiple
 threads will write/read.
try:
    redis.xgroup_create('mystream','myworkergroup',mkstream=True)
except exceptions.ResponseError as e:
    logging.info("Consumer group already exists. Will continue despite the error:
 {}".format(e))
except:
    raise

# Starts 5 writer threads.
for writer_no in range(5):
    writerThread = threading.Thread(target=writeMessage, name='writer-'+str(writer_no),
 args=('mystream',),daemon=True)
    writerThread.start()

# Starts 10 reader threads
for reader_no in range(10):
    readerThread = threading.Thread(target=readMessage, name='reader-'+str(reader_no),
 args=('myworkergroup','mystream',),daemon=True)
    readerThread.daemon = True
    readerThread.start()

# Keep the code running for 30 seconds
time.sleep(30)
```

Pour exécuter le programme, saisissez la commande suivante.

```
python ReadWriteStream.py
```

Tutoriel : Configuration de Lambda pour un accès ElastiCache dans un VPC

Dans ce didacticiel, vous apprendrez à créer un cache ElastiCache sans serveur, à créer une fonction Lambda, puis à tester la fonction Lambda et éventuellement à la nettoyer ensuite.

Rubriques

- Étape 1 : Création d'un cache ElastiCache sans serveur.
- Étape 2 : créer une fonction Lambda pour ElastiCache
- Étape 3 : tester la fonction Lambda avec ElastiCache
- Étape 4 : Nettoyage (facultatif)

Étape 1 : Création d'un cache ElastiCache sans serveur.

Pour créer un cache sans serveur, procédez comme suit.

Étape 1.1 : Création d'un cache sans serveur

Au cours de cette étape, vous créez un cache sans serveur 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 cache sans serveur à l'aide de la ElastiCache console ou de l'API, consultezCréation d'un cache sans serveur Redis OSS.

```
aws elasticache create-serverless-cache \
  --serverless-cache-name cache-01  \
--description "ElastiCache IAM auth application" \
--engine valkey
```

Notez que la valeur du champ Statut est définie sur `CREATING`. La création de votre cache peut prendre une minute. ElastiCache

Étape 1.2 : Copier le point de terminaison du cache sans serveur

Vérifiez que ElastiCache pour Redis, OSS a terminé de créer le cache avec la `describe-serverless-caches` commande.

```
aws elasticache describe-serverless-caches \
--serverless-cache-name cache-01
```

Copiez l'adresse du point de terminaison affichée 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.

   JSON

   ```
   {
   "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.

   JSON

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

```
"Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-
east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

3. Créez un rôle IAM.

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. Créez la politique IAM.

```
aws iam create-policy \
  --policy-name "elasticache-allow-all" \
  --policy-document file://policy.json
```

5. Attachez la politique gérée IAM au rôle.

```
aws iam attach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Étape 1.4 : Création d'un utilisateur par défaut

1. Créez un nouvel utilisateur par défaut.

```
aws elasticache create-user \
  --user-name default \
--user-id default-user-disabled \
--engine redis \
--authentication-mode Type=no-password-required \
--access-string "off +get ~keys*"
```

2. Créez un compte utilisateur prenant en charge IAM.

```
aws elasticache create-user \
  --user-name iam-user-01 \
--user-id iam-user-01 \
--authentication-mode Type=iam \
--engine redis \
--access-string "on ~* +@all"
```

3. Créez un groupe d'utilisateurs et attachez l'utilisateur.

```
aws elasticache create-user-group \
  --user-group-id iam-user-group-01 \
--engine redis \
--user-ids default-user-disabled iam-user-01

aws elasticache modify-serverless-cache \
  --serverless-cache-name cache-01  \
--user-group-id iam-user-group-01
```

Étape 2 : créer une fonction Lambda pour ElastiCache

Pour créer une fonction Lambda permettant d'accéder au ElastiCache cache, procédez comme suit.

Étape 2.1 : Création d'une fonction Lambda

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 ElastiCache cache. Copiez le code et enregistrez-le dans un fichier nommé `app.py`. Assurez-vous de remplacer la `elasticache_endpoint` valeur du code par l'adresse du point de terminaison que vous avez copiée à l'étape précédente.

```
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 ElastiCacheIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cache_name, is_serverless=False, region="us-east-1"):
        self.user = user
        self.cache_name = cache_name
        self.is_serverless = is_serverless
        self.region = region

        session = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "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}
        if self.is_serverless:
            query_params["ResourceType"] = "ServerlessCache"
        url = urlunparse(
            ParseResult(
                scheme="https",
                netloc=self.cache_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
```

```
        # Elasticache 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
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.use1.cache.amazonaws.com" #
 replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
 is_serverless=True)
    redis_client = redis.Redis(host=elasticache_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 cache
    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 cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Valkey.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
 {decoded_result}")

    return "Fetched value from Valkey"
```

Ce code utilise la bibliothèque Python redis-py pour mettre des éléments dans votre cache et les récupérer. Ce code utilise des outils de cache 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 source app.py, créez un package de dossiers dans lequel installer les bibliothèques redis-py et cachetools.

```
mkdir package
```

Installez redis-py, cachetools en utilisant pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Créez un fichier .zip contenant les bibliothèques redis-py et cachetools. Sous Linux et macOS, exécutez la commande suivante. Sous Windows, utilisez l'utilitaire zip de votre choix pour créer un fichier .zip avec les bibliothèques redis-py et cachetools 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 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 "elasticache-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 (AccessValkey) 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 précédente. 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, choisissez Votre VPCs, puis choisissez le VPC par défaut de votre AWS 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 AccessValkey  \
--region us-east-1 \
--zip-file fileb://my_deployment_package.zip \
--role arn:aws:iam::123456789012:role/elasticache-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 avec ElastiCache

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 (AccessValkey) à l'aide de la commande AWS Lambda invoke.

   ```
   aws lambda invoke \
   --function-name AccessValkey  \
   --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 (/aws/lambda/AccessValkey). Le flux de journaux doit contenir un résultat similaire à ce qui suit :

     ```
     Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched
      826e70c5f4d2478c8c18027125a3e01e from Valkey.
     ```

   - Passez en revue les résultats dans la AWS Lambda console.

Étape 4 : Nettoyage (facultatif)

Pour nettoyer, procédez comme suit.

Étape 4.1 : Supprimer la fonction Lambda

```
aws lambda delete-function \
 --function-name AccessValkey
```

Étape 4.2 : Supprimer le cache sans serveur

Supprimez le cache.

```
aws elasticache delete-serverless-cache \
 --serverless-cache-name cache-01
```

Supprimez des utilisateurs et des groupes d'utilisateurs.

```
aws elasticache delete-user \
 --user-id default-user-disabled

aws elasticache delete-user \
 --user-id iam-user-01

aws elasticache delete-user-group \
 --user-group-id iam-user-group-01
```

Étape 4.3 : Supprimer le rôle et les politiques IAM

```
aws iam detach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"

aws iam detach-role-policy \
--role-name "elasticache-iam-auth-app" \
--policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"

aws iam delete-role \
 --role-name "elasticache-iam-auth-app"

 aws iam delete-policy \
```

```
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Conception et gestion de votre propre ElastiCache cluster

Si vous avez besoin d'un contrôle précis de votre ElastiCache cluster, vous pouvez choisir de concevoir votre propre cluster. ElastiCache vous permet d'exploiter un cluster basé sur des nœuds en choisissant le type de nœud, le nombre de nœuds et le placement des nœuds dans les zones de AWS disponibilité de votre cluster. Comme il ElastiCache s'agit d'un service entièrement géré, il gère automatiquement le provisionnement du matériel, la surveillance, le remplacement des nœuds et les correctifs logiciels pour votre cluster.

Pour en savoir plus sur sa configuration, consultez Con ElastiCachefiguration. Pour plus de détails sur la gestion, la mise à jour ou la suppression de nœuds ou de clusters, consultez Gestion des nœuds dans ElastiCache. Pour un aperçu des principaux composants d'un ElastiCache déploiement Amazon lorsque vous concevez votre propre ElastiCache cluster, consultez ces concepts clés.

Rubriques

- ElastiCache composants et fonctionnalités
- ElastiCache terminologie
- Tutoriel : Comment concevoir votre propre cluster
- Suppression d'un cluster
- Autres ElastiCache tutoriels et vidéos
- Gestion des nœuds dans ElastiCache
- Gestion des clusters dans ElastiCache
- Comparaison des caches auto-conçus par Valkey, Memcached et Redis OSS
- Migration en ligne pour Valkey ou Redis OSS
- Choix des régions et des zones de disponibilité pour ElastiCache

ElastiCache composants et fonctionnalités

Vous trouverez ci-dessous un aperçu des principaux composants d'un ElastiCache déploiement Amazon.

Rubriques

- ElastiCache nœuds

- ElastiCache éclats
- ElastiCache clusters
- ElastiCache réplication
- ElastiCache points de terminaison
- ElastiCache groupes de paramètres
- ElastiCache sécurité
- ElastiCache groupes de sous-réseaux
- ElastiCache sauvegardes
- ElastiCache événements

ElastiCache nœuds

Un nœud est le plus petit élément constitutif d'un ElastiCache déploiement. Un nœud peut exister par lui-même ou en relation avec d'autres nœuds.

Un nœud est un bloc de RAM sécurisée, de dimension fixe et attachée à un réseau. Chaque nœud exécute une instance du moteur et de la version choisis quand vous avez créé le cluster. Si nécessaire, vous pouvez faire évoluer les nœuds d'un cluster vers une instance d'un type différent. Pour de plus amples informations, veuillez consulter Dimensionnement ElastiCache.

Chaque nœud d'un cluster est du même type d'instance et exécute le même moteur de cache. Chaque nœud de cache a son propre nom DNS et son propre port. Plusieurs types de nœuds de cache sont pris en charge, chacun avec différentes tailles de mémoire associée. Pour obtenir une liste des types d'instances de nœuds pris en charge, consultez Types de nœuds pris en charge.

Vous pouvez acheter des nœuds sur une pay-as-you-go base où vous ne payez que pour l'utilisation d'un nœud. Ou bien vous pouvez acquérir des nœuds réservés à un coût horaire beaucoup plus avantageux. Si votre taux d'utilisation est élevé, vous pouvez faire des économies en achetant des nœuds réservés. Imaginons que votre cluster est utilisé en permanence et que vous ajoutez des nœuds pour faire face aux pics d'utilisation. Dans ce cas, vous pouvez acheter un certain nombre de nœuds réservés à exécuter la plupart du temps. Vous pouvez ensuite acheter pay-as-you-go des nœuds pour les moments où vous devez parfois ajouter des nœuds. Pour plus d'informations sur les nœuds réservés, consultez Nœuds réservés.

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

ElastiCache éclats

Une partition Valkey ou Redis OSS (appelée groupe de nœuds dans l'API et la CLI) est un regroupement de un à six nœuds connexes. Un cluster Valkey ou Redis OSS avec le mode cluster activé possède toujours au moins une partition.

Le sharding est une méthode de partitionnement de bases de données qui sépare les grandes bases de données en parties plus petites, plus rapides et plus faciles à gérer, appelées fragments de données. Cela peut améliorer l'efficacité de la base de données en répartissant les opérations entre plusieurs sections distinctes. L'utilisation de partitions peut offrir de nombreux avantages, notamment une amélioration des performances, de l'évolutivité et de la rentabilité.

Les clusters Valkey et Redis OSS avec le mode cluster activé peuvent contenir jusqu'à 500 partitions, vos données étant partitionnées entre les partitions. La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur Valkey ou Redis OSS est 5.0.6 ou supérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux. Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Une partition avec plusieurs nœuds implémente la réplication avec un nœud principal en lecture/écriture et de 1 à 5 nœuds de réplica. Pour de plus amples informations, veuillez consulter Haute disponibilité avec les groupes de réplication.

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

ElastiCache clusters

Un cluster est un regroupement logique d'un ou de plusieurs nœuds. Les données sont partitionnées entre les nœuds d'un cluster Memcached et entre les partitions d'un cluster Valkey ou Redis OSS dont le mode cluster est activé.

De nombreuses ElastiCache opérations ciblent les clusters :

- Création d'un cluster
- Modification d'un cluster
- Réalisation d'instantanés d'un cluster (toutes les versions de Redis)
- 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 dans ElastiCache et Gestion des nœuds dans ElastiCache

  Informations sur les clusters, les nœuds et les opérations connexes.
- AWS limites de service : Amazon ElastiCache

  Informations sur ElastiCache les limites, telles que le nombre maximum de nœuds ou de clusters. Pour dépasser certaines de ces limites, vous pouvez faire une demande à l'aide du formulaire de demande de nœud de ElastiCache cache Amazon.
- Atténuation des défaillances

  Informations sur l'amélioration de la tolérance aux pannes de vos clusters et de vos groupes de réplication Valkey ou Redis OSS.

Configurations de clusters classiques

Voici des configuration de cluster classiques.

Clusters Valkey ou Redis OSS

Les clusters Valkey ou Redis OSS dont le mode cluster est désactivé contiennent toujours une seule partition (dans l'API et la CLI, un groupe de nœuds). Une partition Valkey ou Redis OSS contient un à six nœuds. S'il y a plus d'un nœud dans une partition, celle-ci prend en charge la réplication. Dans ce cas, un nœud est le nœud read/write principal et les autres sont des nœuds de réplication en lecture seule.

Pour une meilleure tolérance aux pannes, nous recommandons d'avoir au moins deux nœuds dans un cluster Valkey ou Redis OSS et d'activer le mode multi-AZ. Pour de plus amples informations, veuillez consulter Atténuation des défaillances.

À mesure que la demande de votre cluster Valkey ou Redis OSS change, vous pouvez augmenter ou diminuer. Pour ce faire, déplacez votre cluster vers un autre type d'instance de nœud. Si votre application est gourmande en lecture, nous vous recommandons d'ajouter des répliques en lecture seule au cluster. En faisant cela, vous pouvez répartir les lectures sur un nombre plus approprié de nœuds.

Vous pouvez également utiliser la hiérarchisation des données. Les données les plus fréquemment consultées sont stockées en mémoire et les données les moins fréquemment consultées sont stockées sur disque. L'avantage de l'utilisation de la hiérarchisation des données est qu'elle réduit les besoins en mémoire. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

ElastiCache permet de changer dynamiquement le type de nœud d'un cluster Valkey ou Redis OSS en un type de nœud plus grand. Pour plus d'informations sur le dimensionnement, consultez Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé) ou Dimensionnement des nœuds de réplication pour Valkey ou Redis OSS (mode cluster désactivé).

Configurations de cluster typiques pour Memcached

Memcached prend en charge jusqu'à 300 nœuds par client pour chaque AWS région, chaque cluster comportant de 1 à 60 nœuds. Vous partitionnez vos données sur plusieurs nœuds dans un cluster Memcached.

Lorsque vous exécutez le moteur Memcached, les clusters peuvent être composés de 1 à 60 nœuds. Vous partitionnez votre base de données sur plusieurs nœuds. Votre application a un accès en lecture et en écriture sur le point de terminaison de chaque nœud. Pour plus d'informations, consultez Découverte automatique.

Pour améliorer la tolérance aux pannes, localisez vos nœuds Memcached dans différentes zones de disponibilité (AZs) au sein de la région du AWS cluster. Ainsi, une défaillance dans une zone de disponibilité a un impact minimal sur l'ensemble du cluster et de l'application. Pour de plus amples informations, veuillez consulter Atténuation des défaillances.

Au fur et à mesure que votre cluster Memcached change, vous pouvez le faire évoluer en ajoutant ou supprimant des nœuds, qui répartissent vos données sur le nouvel ensemble de nœuds. Lorsque vous partitionnez vos données, nous vous recommandons d'utiliser le hachage cohérent. Pour plus d'informations sur le hachage cohérent, consultez Configuration de votre ElastiCache client pour un équilibrage de charge efficace (Memcached).

ElastiCache réplication

Pour Valkey et Redis OSS, la réplication est mise en œuvre en regroupant de deux à six nœuds dans un shard (dans l'API et la CLI, appelé groupe de nœuds). L'un de ces nœuds est le nœud principal en lecture/écriture. Tous les autres nœuds sont des nœuds de réplica en lecture seule. Les réplications ne sont disponibles que ElastiCache pour Valkey et Redis OSS, et non pour ElastiCache Memcached.

Chaque nœud de réplica conserve une copie des données du nœud principal. Les nœuds de réplica utilisent des mécanismes de réplication asynchrones pour maintenir les réplicas en lecture synchronisés avec le nœud principal. Les applications peuvent lire à partir de n'importe lequel des nœuds du cluster, mais peuvent écrire uniquement dans le cluster principal. Les réplicas en lecture améliorent l'adaptabilité en répartissant les lectures sur plusieurs points de terminaison. Les réplicas en lecture améliorent également la tolérance aux pannes en conservant plusieurs copies des données de Le fait de répartir les réplicas en lecture sur plusieurs Zones de disponibilité permet d'améliorer davantage la tolérance aux pannes. Pour plus d'informations sur la tolérance aux pannes, consultez Atténuation des défaillances.

Les clusters Valkey ou Redis OSS prennent en charge une partition (dans l'API et la CLI, appelée groupe de nœuds).

La réplication du point de vue de l'API et de la CLI utilise une terminologie différente pour assurer la compatibilité avec les versions antérieures, mais les résultats sont identiques. Le tableau suivant présente les termes employés par l'API et la CLI pour l'implémentation de la réplication.

Comparaison de la réplication : Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé) --> Cluster Valkey ou Redis OSS avec le mode cluster activé par rapport au cluster Valkey ou Redis OSS avec le mode cluster désactivé

Dans le tableau suivant, vous trouverez une comparaison des fonctionnalités des groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) et des groupes de réplication Valkey ou Redis OSS (mode cluster activé).

	Cluster Valkey ou Redis OSS avec le mode cluster désactivé	Cluster Valkey ou Redis OSS avec le mode cluster activé
Partitions (groupes de nœuds)	1	1–500

	Cluster Valkey ou Redis OSS avec le mode cluster désactivé	Cluster Valkey ou Redis OSS avec le mode cluster activé
Répliques pour chaque partition (groupe de nœuds)	0–5	0–5
Partitionnement de données	Non	Oui
Ajouter/supprimer des réplicas	Oui	Oui
Ajouter/supprimer des groupes de nœuds	Non	Oui
Prend en charge l'augment ation d'échelle	Oui	Oui
Prend en charge les mises à niveau du moteur	Oui	Oui
Promouvoir le réplica en primaire	Oui	Automatique
Multi-AZ	Facultatif	Obligatoire
Sauvegarder/restaurer	Oui	Oui
Remarques :		
Si un primaire n'a pas de réplicas et qu'il échoue, vous perdez toutes les données de ce primaire.		
Vous pouvez utiliser la sauvegarde et la restauration pour migrer vers Valkey ou Redis OSS (mode cluster activé).		
Vous pouvez utiliser la sauvegarde et la restauration pour redimensionner votre cluster Valkey ou Redis OSS (mode cluster activé).		

Toutes les partitions (dans l'API et la CLI, groupes de nœuds) et tous les nœuds doivent résider dans la même région AWS . Cependant, vous pouvez approvisionner les nœuds individuels dans plusieurs zones de disponibilité au sein de cette AWS région.

Les réplicas en lecture évitent les pertes de données potentielles car les données sont répliquées sur deux ou plusieurs nœuds, le principal et un ou plusieurs réplicas en lecture. Pour une plus grande fiabilité et une récupération plus rapide, nous vous recommandons de créer une ou plusieurs répliques de lecture dans différentes zones de disponibilité.

Vous pouvez également tirer parti des banques de données mondiales. En utilisant la fonctionnalité Global Datastore pour Redis OSS, vous pouvez utiliser une réplication entièrement gérée, rapide, fiable et sécurisée entre les régions AWS . Grâce à cette fonctionnalité, vous pouvez créer des clusters de répliques de lecture entre régions afin de permettre des lectures ElastiCache à faible latence et une reprise après sinistre dans toutes les régions AWS . Pour plus d'informations, voir Réplication entre AWS régions à l'aide de banques de données globales.

Réplication : limites et exclusions

- Multi-AZ n'est pas pris en charge sur les types de nœuds T1.

ElastiCache points de terminaison

Un point de terminaison est l'adresse unique que votre application utilise pour se connecter à un ElastiCache nœud ou à un cluster.

Points de terminaison à nœud unique pour Valkey ou Redis OSS avec le mode cluster désactivé

Le point de terminaison d'un cluster Valkey ou Redis OSS à nœud unique est utilisé pour se connecter au cluster pour les lectures et les écritures.

Points de terminaison à nœuds multiples pour Valkey ou Redis OSS avec le mode cluster désactivé

Un cluster Valkey ou Redis OSS à plusieurs nœuds dont le mode cluster est désactivé possède deux types de points de terminaison. Le point de terminaison principal se connecte toujours au nœud principal du cluster, même si le rôle du nœud spécifique dans le principal change. Utilisez le point de terminaison principal pour toutes les écritures dans le cluster.

Utilisez le Point de terminaison du lecteur pour répartir également les connexions entrantes vers le point de terminaison entre toutes les réplicas lues. Utilisez les points de terminaison de nœud individuels pour les opérations de lecture (ils API/CLI sont appelés points de terminaison de lecture).

Points de terminaison Valkey ou Redis OSS (mode cluster activé)

Un cluster Valkey ou Redis OSS avec le mode cluster activé possède un point de terminaison de configuration unique. En se connectant au point de terminaison de configuration, votre application est en mesure de découvrir les points de terminaison principal et de lecture pour chaque partition du cluster.

Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

ElastiCache pour les points de terminaison Memcached

Chaque nœud d'un cluster Memcached a son propre point de terminaison. Le cluster a également un point de terminaison appelé le configuration endpoint (point de terminaison de la configuration). Si vous activez La découverte automatique et que vous vous connectez au point de terminaison de configuration, votre application découvre automatiquement le point de terminaison de chaque nœud, même après l'ajout ou la suppression des nœuds du cluster. Pour plus d'informations, consultez Découverte automatique.

Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

ElastiCache groupes de paramètres

Les groupes de paramètres de cache sont un moyen simple de gérer les paramètres d'exécution pour le logiciel de moteur pris en charge. Les paramètres permettent de contrôler l'utilisation de la mémoire, les règles d'expulsion, la taille des objets, etc. Un groupe de ElastiCache paramètres est un ensemble nommé de paramètres spécifiques au moteur que vous pouvez appliquer à un cluster. En faisant cela, vous vous assurez que tous les nœuds de ce cluster sont configurés exactement de la même manière.

Pour une liste des paramètres pris en charge, leurs valeurs par défaut et ceux qui peuvent être modifiés, consultez DescribeEngineDefaultParameters (CLI : describe-engine-default-parameters).

Pour des informations plus détaillées sur les groupes de ElastiCache paramètres, consultezConfiguration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres.

ElastiCache sécurité

Pour renforcer la sécurité, l'accès aux ElastiCache nœuds est limité aux applications exécutées sur les EC2 instances Amazon que vous autorisez. Vous pouvez contrôler les EC2 instances Amazon qui peuvent accéder à votre cluster à l'aide de groupes de sécurité.

Par défaut, tous les nouveaux ElastiCache clusters sont lancés dans un environnement Amazon Virtual Private Cloud (Amazon VPC). Vous pouvez utiliser des groupes de sous-réseaux pour accorder l'accès au cluster à partir d' EC2instances Amazon exécutées sur des sous-réseaux spécifiques.

Outre la restriction de l'accès aux nœuds, ElastiCache prend en charge le protocole TLS et le chiffrement sur place pour les nœuds exécutant des versions spécifiées de. ElastiCache Pour plus d'informations, consultez les ressources suivantes :

- Sécurité des données sur Amazon ElastiCache
- Authentification avec la commande Valkey et Redis OSS AUTH

ElastiCache 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 de VPC Amazon.

Si vous créez un cluster dans un Amazon VPC, vous devez spécifier un groupe de sous-réseaux. ElastiCache utilise ce groupe de sous-réseaux de cache pour choisir un sous-réseau et les adresses IP au sein de ce sous-réseau à associer à vos nœuds de cache.

Pour plus d'informations sur l'utilisation du groupe de sous-réseaux de cache dans un environnement Amazon VPC, consultez les ressources suivantes :

- Amazon VPCs et la ElastiCache sécurité
- Étape 3. Autoriser l'accès au cluster
- Sous-réseaux et groupes de sous-réseaux

ElastiCache sauvegardes

Une sauvegarde est une point-in-time copie d'un cluster Valkey ou Redis OSS ou d'un cache sans serveur, ou d'un cache sans serveur Memcached. Les sauvegardes peuvent être utilisées pour

restaurer un cluster existant ou pour amorcer un nouveau cluster. Les sauvegardes sont constituées de toutes les données d'un cluster, plus quelques métadonnées.

Selon la version de Valkey ou Redis OSS exécutée sur votre cluster, le processus de sauvegarde nécessite différentes quantités de mémoire réservée pour réussir. Pour plus d'informations, consultez les ressources suivantes :

- Instantané et restauration
- Implémentation de la sauvegarde et de la synchronisation
- Impact sur les performances des sauvegardes de clusters auto-conçus
- S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS

ElastiCache événements

Lorsque des événements importants se produisent sur un cluster de cache, ElastiCache envoie une notification à une rubrique Amazon SNS spécifique. Ces événements peuvent inclure des éléments tels que l'échec ou la réussite de l'ajout d'un nœud, une modification du groupe de sécurité, etc. En surveillant les événements clés, vous pouvez connaître l'état actuel de vos clusters et, dans de nombreux cas, prendre des actions correctives.

Pour plus d'informations sur ElastiCache les événements, consultezSurveillance des événements par Amazon SNS ElastiCache .

ElastiCache terminologie

En octobre 2016, Amazon ElastiCache a lancé le support pour Redis OSS 3.2. À ce stade, nous avons ajouté la possibilité de partitionner vos données sur un maximum de 500 partitions (appelées groupes de nœuds dans l' ElastiCache API et AWS CLI). Pour préserver la compatibilité avec les versions précédentes, nous avons étendu les opérations de la version 2015-02-02 de l'API pour inclure la nouvelle fonctionnalité Redis OSS.

Dans le même temps, nous avons commencé à utiliser la terminologie de la ElastiCache console qui est utilisée dans cette nouvelle fonctionnalité et qui est courante dans l'ensemble du secteur. Ces modifications impliquent que la terminologie utilisée dans l'API et la CLI est parfois différente de celle utilisée dans la console. La liste suivante identifie les termes qui peuvent varier entre l'API, la CLI et la console.

Cluster ou nœud de cache par rapport au nœud

Il existe une one-to-one relation entre un nœud et un cluster de cache lorsqu'il n'existe aucun nœud de réplication. Ainsi, la ElastiCache console utilisait souvent les termes de manière interchangeable. La console utilise désormais le terme nœud partout. La seule exception est le bouton Créer un cluster, qui lance le processus de création d'un cluster avec ou sans nœuds de réplica.

L' ElastiCache API et AWS CLI continuez à utiliser les termes comme par le passé.

Cluster ou groupe de réplication Valkey ou Redis OSS

La console utilise désormais le terme cluster pour tous ElastiCache pour les clusters Redis OSS. La console utilise le terme cluster dans tous les cas de figure suivants :

- Lorsque le cluster est un cluster Valkey ou Redis OSS à nœud unique.
- Lorsque le cluster est un cluster Valkey ou Redis OSS (mode cluster désactivé) qui prend en charge la réplication au sein d'une seule partition (dans l'API et la CLI, appelée groupe de nœuds).
- Lorsque le cluster est un cluster Valkey ou Redis OSS (mode cluster activé) qui prend en charge la réplication entre 1 et 90 partitions ou jusqu'à 500 avec une demande d'augmentation de limite. Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Pour plus d'informations sur les groupes de réplication Valkey ou Redis OSS, consultez. Haute disponibilité avec les groupes de réplication

Le schéma suivant illustre les différentes topologies des ElastiCache clusters Redis OSS du point de vue de la console.

L' ElastiCache API et les AWS CLI opérations distinguent toujours un nœud unique ElastiCache pour les clusters Redis OSS des groupes de réplication Valkey ou Redis OSS à nœuds multiples. Le schéma suivant illustre les différentes ElastiCache topologies de Redis OSS du point de vue de l' ElastiCache API et AWS CLI du point de vue.

Groupe de réplication Valkey ou Redis OSS par rapport à une banque de données globale

Une banque de données globale est un ensemble d'un ou de plusieurs clusters qui se répliquent entre eux dans différentes régions, tandis qu'un groupe de réplication Valkey ou Redis OSS réplique les données sur un cluster activé en mode cluster avec plusieurs partitions. Un magasin de données global se compose des éléments suivants :

- Cluster principal (actif) : un cluster principal accepte les écritures répliquées dans tous les clusters de l'entrepôt de données global. Un cluster principal accepte également les demandes de lecture.
- Cluster secondaire (passif) : un cluster secondaire accepte uniquement les demandes de lecture et réplique les mises à jour de données à partir d'un cluster principal. Un cluster secondaire doit se trouver dans une AWS région différente de celle du cluster principal.

Pour plus d'informations sur les entrepôts de données globales, consultez Réplication entre AWS régions à l'aide de banques de données mondiales.

Tutoriel : Comment concevoir votre propre cluster

Voici comment concevoir votre propre cluster pour Valkey, Memcached et Redis OSS.

Rubriques

- Conception de votre propre ElastiCache cluster (Valkey)
- Conception de votre propre cluster ElastiCache Redis OSS

Conception de votre propre ElastiCache cluster (Valkey)

Voici les actions ponctuelles que vous devez effectuer pour commencer à concevoir votre cluster ElastiCache (Valkey).

Étape 1 : Créer un groupe de sous-réseaux

Avant de créer un cluster ElastiCache (Valkey), vous devez d'abord créer un groupe de sous-réseaux. Un groupe de sous-réseaux de cache est un ensemble de sous-réseaux que vous souhaitez indiquer dans vos clusters de cache dans un VPC. Lors du lancement d'un cluster de cache dans un VPC, vous devez sélectionner un groupe de sous-réseaux de cache. ElastiCache Utilise ensuite ce groupe de sous-réseaux de cache pour attribuer des adresses IP au sein de ce sous-réseau à chaque nœud de cache du cluster.

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.

Pour plus d'informations sur la configuration, ElastiCache voirCon ElastiCachefiguration.

Les procédures suivantes expliquent comment créer un groupe de sous-réseaux appelé `mysubnetgroup` à l'aide de la console et de l' AWS CLI.

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 AWS de gestion et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste de navigation, sélectionnez Subnet Groups.
3. Choisissez Créer un groupe de sous-réseaux.
4. Dans l'assistant Create Subnet Group, procédez comme suit. Lorsque tous les paramètres sont définis selon vos choix, choisissez Yes, Create.
   a. Dans le champ Name, saisissez le nom de votre groupe de sous-réseaux de
   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.
   d. Dans les listes de zone de disponibilité et d'ID de sous-réseau, choisissez la zone de disponibilité ou Utilisation de zones locales avec ElastiCache l'ID de votre sous-réseau privé, puis choisissez Ajouter.

5. 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 ElastiCache console. 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.

Créer un groupe de sous-réseaux (AWS CLI)

A l'invite de commande, utilisez la commande `create-cache-subnet-group` pour créer un groupe de sous-réseaux de

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
```

```
    --subnet-ids subnet-53df9c3a
```

Pour Windows :

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubriquecreate-cache-subnet-group.

Étape 2 : 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 ElastiCache. Pour les besoins de cet exercice de mise en route, vous allez créer un cluster avec le mode cluster désactivé et 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). Les frais d' ElastiCache utilisation standard de l'instance vous seront facturés 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 ElastiCache d'utilisation, consultez Amazon ElastiCache.

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

Création d'un cluster Valkey (mode cluster désactivé) (console)

Pour créer un cluster Valkey (mode cluster désactivé) à l'aide de la console ElastiCache

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région dans laquelle vous souhaitez lancer ce cluster.
3. Choisissez Get started (Mise en route) dans le panneau de navigation.
4. Choisissez Create VPC (Créer un VPC) et suivez les étapes décrites à la page Création d'un cloud privé virtuel (VPC).
5. Sur la page du tableau de ElastiCache bord, choisissez Valkey cache ou Redis OSS cache, puis choisissez Create Valkey cache ou Create Redis OSS cache.
6. Sous Cluster settings (Paramètres du cluster), procédez comme suit :
   a. Choisissez Configure and create a new cluster (Configurer et créer un cluster).
   b. Pour Cluster mode (Mode du cluster), choisissez Disabled (Désactivé).
   c. Pour Cluster info (Informations sur le cluster), saisissez une valeur pour Name (Nom).
   d. (Facultatif) Saisissez une valeur pour Description.
7. Sous Location (Emplacement) :

   AWS Cloud

   1. Pour AWS Cloud, nous vous recommandons d'accepter les paramètres par défaut pour Multi-AZ et Auto-failover (Basculement automatique). Pour plus d'informations, consultez Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ.
   2. Sous Cluster settings (Paramètres de cluster)
      a. Pour Engine version (Version du moteur), choisissez une version disponible.
      b. Pour Port, utilisez le port par défaut, 6379. Si vous avez une raison d'utiliser un autre port, saisissez le numéro de port.

c. Pour Groupe de paramètres, choisissez un groupe de paramètres ou créez-en un nouveau. Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres OSS Valkey et Redis et Création d'un groupe ElastiCache de paramètres.

> **Note**
>
> Lorsque vous sélectionnez un groupe de paramètres pour définir les valeurs de configuration du moteur, ce groupe de paramètres est appliqué à tous les clusters du magasin de données global. Dans la page Parameter Groups (Groupes de paramètres) l'attribut Global yes/no (oui/non) indique si un groupe de paramètres fait partie d'un magasin de données global.

d. Pour Type de nœud, choisissez la flèche vers le bas ( ). Dans la boîte de dialogue Modifier le type de nœud choisissez une valeur pour la famille d'instances pour le type de nœud souhaité. Choisissez ensuite le type de nœud que vous souhaitez utiliser pour ce cluster, puis choisissez Enregistrer.

   Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

   Si vous choisissez un type de nœud r6gd, la hiérarchisation des données est automatiquement activée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

e. Pour Number of replicas (Nombre de réplicas), choisissez le nombre de réplicas en lecture que vous voulez. Si vous avez activé Multi-AZ, le nombre doit être compris entre 1 et 5.

3. Sous Connectivity (Connectivité)

   a. Dans Network type (Type de réseau), choisissez la ou les versions d'adresses IP que ce cluster prendra en charge.

   b. Pour les groupes de sous-réseaux, choisissez le sous-réseau que vous souhaitez appliquer à ce cluster. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds. ElastiCache les clusters nécessitent un sous-réseau à double pile auquel les deux

IPv6 adresses sont attribuées pour fonctionner en mode double pile IPv4 et un sous-réseau « uniquement » pour fonctionner en tant que « IPv6 -only ». IPv6

Lorsque vous créez un nouveau groupe de sous-réseaux, entrez le VPC ID (ID du VPC) auquel il appartient.

Pour plus d'informations, consultez :

- Choix d'un type de réseau dans ElastiCache.
- Créer un sous-réseau dans votre VPC

Si vous utilisez Utilisation de zones locales avec ElastiCache, vous devez créer ou choisir un sous-réseau qui se trouve dans la zone locale.

Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux.

4. Pour Availability zone placements (Placement de zones de disponibilité), vous avez deux options :

   - Aucune préférence : ElastiCache choisit la zone de disponibilité.
   - Specify availability zones (Spécifier les zones de disponibilité) : vous spécifiez la zone de disponibilité pour chaque cluster.

     Si vous avez choisi de spécifier les Zones de disponibilité, pour chaque cluster de chaque partition, choisissez la Zone de disponibilité depuis la liste.

   Pour de plus amples informations, veuillez consulter Choix des régions et des zones de disponibilité pour ElastiCache.

5. Choisissez Next (Suivant)
6. Dans les paramètres avancés de Valkey ou Redis OSS

   - Pour Security (Sécurité) :

     i. 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 autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour de plus amples informations, veuillez consulter Utilisation clés AWS KMS gérées par le client.

- Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour Valkey et pour Redis OSS 6.0 et versions ultérieures, si vous activez le chiffrement en transit, vous serez invité à spécifier l'une des options de contrôle d'accès suivantes :
  - No Access Control (Aucun contrôle d'accès) – il s'agit du paramètre par défaut. Cela indique qu'aucune restriction n'est imposée à l'accès des utilisateurs au cluster.
  - User Group Access Control List (Liste de contrôle d'accès au groupe d'utilisateurs) : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs pouvant accéder au cluster. Pour de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.
  - Utilisateur par défaut AUTH — Mécanisme d'authentification pour le serveur Redis OSS. Pour plus d'informations, consultez AUTH.
- AUTH — Mécanisme d'authentification pour le serveur Redis OSS. Pour plus d'informations, consultez AUTH.

> **Note**
>
> Pour les versions de Valkey et Redis OSS supérieures à 3.2.6, à l'exception de la version 3.2.10, Redis OSS AUTH est la seule option.

ii. 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.

7. Pour des sauvegardes automatiques régulières, choisissez Activer les sauvegardes automatiques, puis entrez le nombre de jours pendant lesquels vous souhaitez conserver une sauvegarde automatique avant sa suppression automatique. Si vous ne souhaitez pas de sauvegardes automatiques régulières, désactivez la case à cocher Enable automatic backups. Dans les deux cas, vous avez toujours la possibilité de créer des sauvegardes manuelles.

   Pour plus d'informations sur la sauvegarde et la restauration de Redis OSS, consultezInstantané et restauration.

8. (Facultatif) Spécifiez une fenêtre de maintenance. La fenêtre de maintenance indique le moment, généralement d'une durée d'une heure, chaque semaine, où ElastiCache planifie la maintenance du système pour votre cluster. Vous pouvez autoriser ElastiCache à choisir le jour et l'heure de la fenêtre de maintenance (Aucune préférence) ou vous pouvez choisir vous-même le jour, l'heure et la durée (Specify maintenance window (Spécifier 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 de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

9. (Facultatif) Pour Logs (Journaux) :

   - Sous Log format (Format de journal), sélectionnez Text (Texte) ou JSON.
   - Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
   - Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatch journaux ou le nom de votre flux Firehose, ou choisissez Select existing, puis choisissez le nom de votre groupe de journaux de CloudWatch journaux ou le nom de votre stream Firehose,

10. Pour les balises, pour vous aider à gérer vos clusters et autres ElastiCache ressources, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Pour plus d'informations, consultez Marquer vos ressources ElastiCache .

11. Choisissez Suivant.

12. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create (Créer).

On premises

1. Pour On premises (Sur site), nous vous recommandons de laisser Auto-failover (Basculement automatique) activé. Pour plus d'informations, voir Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ
2. Pour terminer la création du cluster, suivez les étapes de la section Utilisation d'Outposts.

Dès que le statut de votre cluster est disponible, vous pouvez autoriser Amazon à y EC2 accéder, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 3. Autoriser l'accès au cluster et Étape 4 : Connectez-vous au nœud du cluster.

> **Important**
>
> Une fois que votre cluster est disponible, vous êtes facturé pour chaque heure ou partie d'heure 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. Consultez Supprimer un cluster dans ElastiCache.

Pour utiliser le mode cluster activé, veuillez consulter les rubriques suivantes :

- Pour utiliser la console, veuillez consulter Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console).
- Pour utiliser le AWS CLI, voirCréation d'un cluster Valkey ou Redis OSS (mode cluster activé) ()AWS CLI.

Étape 3. Autoriser l'accès au cluster

Cette section part du principe que vous êtes familiarisé avec le lancement et la connexion aux EC2 instances Amazon. Pour plus d'informations, consultez le guide de EC2 démarrage Amazon.

Tous les ElastiCache clusters sont conçus pour être accessibles depuis une EC2 instance Amazon. Le scénario le plus courant consiste à accéder à un ElastiCache cluster depuis une EC2 instance Amazon dans le même Amazon Virtual Private Cloud (Amazon VPC), ce qui sera le cas pour cet exercice.

Par défaut, l'accès réseau à votre cluster est limité au compte qui a été utilisé pour le créer. Avant de pouvoir vous connecter à un cluster à partir d'une EC2 instance, vous devez autoriser l' EC2 instance à accéder au cluster.

Le cas d'utilisation le plus courant est celui où une application déployée sur une EC2 instance doit se connecter à un cluster dans le même VPC. La méthode la plus simple pour gérer l'accès entre les EC2 instances et les clusters d'un même VPC consiste à effectuer les opérations suivantes :

1. Créez un groupe de sécurité VPC pour votre cluster. Ce groupe de sécurité peut être utilisé pour restreindre l'accès aux instances de cluster. 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 et les groupes de réplication Valkey ou Redis OSS est. `6379`

   > **Important**
   >
   > Les groupes ElastiCache de sécurité Amazon ne s'appliquent qu'aux clusters qui ne s'exécutent pas dans un environnement Amazon Virtual Private Cloud (VPC). Si vous exécutez un Amazon Virtual Private Cloud, les groupes de sécurité ne sont pas disponibles dans le panneau de navigation de la console.
   > Si vous exécutez vos ElastiCache nœuds dans un Amazon VPC, vous contrôlez l'accès à vos clusters à l'aide des groupes de sécurité Amazon VPC, qui sont différents des groupes de sécurité. ElastiCache Pour plus d'informations sur l'utilisation ElastiCache dans un Amazon VPC, consultez Amazon VPCs et la ElastiCache sécurité

2. Créez un groupe de sécurité VPC pour vos EC2 instances (serveurs Web et d'applications). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l' EC2 instance depuis 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' EC2 instance via 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 EC2 instances. N'importe quel membre du groupe de sécurité peut ainsi accéder aux clusters.

> **Note**
>
> Si vous envisagez d'utiliser Local Zones, vérifiez que vous les avez activées. Lorsque vous créez un groupe de sous-réseaux dans cette zone locale, votre VPC est étendu à cette zone locale et votre VPC traitera le sous-réseau comme n'importe quel sous-réseau dans n'importe quelle autre zone de disponibilité. Toutes les passerelles et les tables de routage pertinentes seront automatiquement ajustées.

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 https://console.aws.amazon.com l'adresse /vpc.
2. Dans le panneau de navigation, choisissez Groupes de sécurité.
3. Sélectionnez ou créez un groupe de sécurité que vous utiliserez pour vos instances cluster. 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 et les groupes de réplication Valkey ou Redis OSS 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 EC2 instances Amazon.
5. Choisissez Enregistrer lorsque vous avez terminé.

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

Pour plus d'informations sur l'accès à votre ElastiCache cluster 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 ElastiCache cache dans un Amazon VPC
- Accès aux ElastiCache ressources depuis l'extérieur AWS

Étape 4 : Connectez-vous au nœud du cluster

Avant de continuer, terminez la section Étape 3. Autoriser l'accès au cluster.

Cette section part du principe que vous avez créé une EC2 instance Amazon et que vous pouvez vous y connecter. Pour obtenir des instructions sur la procédure à suivre, consultez le guide de EC2 démarrage Amazon.

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

Rechercher vos points de terminaison de nœud

Lorsque votre cluster est disponible et que vous y avez un accès autorisé, vous pouvez vous connecter à une EC2 instance Amazon et vous connecter au cluster. Pour cela, vous devez d'abord déterminer le point de terminaison.

Trouver les points de terminaison d'un cluster Valkey (mode cluster désactivé) (console)

Si un cluster Valkey (mode cluster désactivé) ne possède qu'un seul nœud, le point de terminaison du nœud est utilisé à la fois pour les lectures et les écritures. Si un cluster a plusieurs nœuds, il y a trois types de points de terminaison : le point de terminaison principal, le point de terminaison du lecteur et les points de terminaison des nœuds.

Le point de terminaison principal est un nom DNS qui est toujours résolu dans le nœud principal du cluster. Le point de terminaison principal n'est pas affecté par les changements apportés à votre cluster, tels que la promotion d'un réplica en lecture au rang de réplica principal. Pour l'activité d'écriture, il est préférable que vos applications se connectent au point de terminaison principal.

Un point de terminaison de lecteur répartira de manière égale les connexions entrantes vers le point de terminaison entre toutes les répliques de lecture d'un ElastiCache cluster. Des facteurs supplémentaires, comme le moment où l'application crée les connexions ou la façon dont l'application utilise ou réutilise les connexions, détermineront la distribution du trafic. Les points de terminaison du lecteur suivent les changements de cluster en temps réel au fur et à mesure que des réplicas sont ajoutés ou supprimés. Vous pouvez placer les multiples répliques de lecture de votre ElastiCache cluster dans différentes zones de AWS disponibilité (AZ) pour garantir la haute disponibilité des points de terminaison des lecteurs.

> **Note**
>
> Un point de terminaison du lecteur n'est pas un équilibreur de charge. Il s'agit d'un enregistrement DNS qui se résoudra à une adresse IP de l'un des nœuds de réplica en mode round robin.

Pour l'activité de lecture, les applications peuvent également se connecter à n'importe quel nœud du cluster. Contrairement au point de terminaison principal, les points de terminaison des nœuds sont résolus dans des points de terminaison spécifiques. En cas de modification dans votre cluster, comme l'ajout ou la suppression d'un réplica, vous devez mettre à jour les points de terminaison des nœuds de votre application.

Pour trouver les points de terminaison d'un cluster Valkey (mode cluster désactivé)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez les caches Valkey ou les caches Redis OSS.

   L'écran des clusters apparaîtra avec une liste qui inclura tous les caches sans serveur Valkey ou Redis OSS existants, les clusters Valkey (mode cluster désactivé) et Valkey (mode cluster activé). Choisissez le cluster que vous avez créé dans la section Création d'un cluster Valkey (mode cluster désactivé) (console).
3. Pour trouver les points de terminaison du and/or lecteur principal du cluster, choisissez le nom du cluster (et non le bouton radio).

Points de terminaison principaux et lecteurs pour un cluster Valkey (mode cluster désactivé)

S'il y a un seul nœud dans le cluster, il n'y a aucun point de terminaison principal et vous pouvez passer à l'étape suivante.

4. Si le cluster Valkey (mode cluster désactivé) possède des nœuds de réplication, vous pouvez trouver les points de terminaison des nœuds de réplication du cluster en choisissant le nom du cluster, puis en choisissant l'onglet Nœuds.

   L'écran des nœuds s'affiche avec chaque nœud du cluster, principal et de réplica, répertorié avec son point de terminaison.

Points de terminaison de nœuds pour un cluster Valkey (mode cluster désactivé)

5. Pour copier un point de terminaison dans le presse-papiers :

   a. Un point de terminaison à la fois, recherchez le point de terminaison que vous souhaitez copier.

   b. Cliquez sur l'icône de copie située juste devant le point de terminaison.

Le point de terminaison est maintenant copié dans votre presse-papiers. Pour en savoir plus sur l'utilisation du point de terminaison pour se connecter à un nœud, consultez Connexion aux nœuds.

Un point de terminaison principal Valkey (mode cluster désactivé) ressemble à ce qui suit. Il existe une différence selon que le chiffrement en transit est activé ou pas.

Chiffrement en transit non activé

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Chiffrement en transit activé

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Pour en savoir plus sur la recherche de vos points de terminaison, consultez les rubriques correspondant au moteur et au type de cluster que vous exécutez.

- Recherche de points de terminaison de connexion dans ElastiCache
- Recherche de points de terminaison pour un cluster Valkey ou Redis OSS (mode cluster activé) (console) – vous avez besoin du point de terminaison de la configuration du cluster.
- Recherche des points de terminaison (AWS CLI)
- Recherche de points de terminaison (ElastiCache API)

Connectez-vous à un cluster ou à un groupe de réplication Valkey ou Redis OSS (Linux)

Maintenant que vous disposez du point de terminaison dont vous avez besoin, vous pouvez vous connecter à une EC2 instance et vous connecter au cluster ou au groupe de réplication. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La dernière version de valkey-cli permet également de connecter SSL/TLS encryption/authentication des clusters activés.

L'exemple suivant utilise des EC2 instances Amazon exécutant Amazon Linux et Amazon Linux 2. Pour plus de détails sur l'installation et la compilation de valkey-cli avec d'autres distributions Linux, consultez la documentation de votre système d'exploitation spécifique.

> **Note**
>
> Ce processus couvre le test d'une connexion à l'aide de l'utilitaire valkey-cli pour une utilisation imprévue uniquement. Pour obtenir la liste des clients Valkey et Redis OSS pris en charge, consultez la documentation de Valkey. Pour des exemples d'utilisation du AWS SDKs with ElastiCache, voirTutoriels : Débuter avec Python et ElastiCache.

Connexion à un mode cluster désactivé cluster non chiffré

1. Exécutez la commande suivante pour vous connecter au cluster et remplacez *primary-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey ou Redis OSS est 6379.)

   ```
   src/valkey-cli -h primary-endpoint -p port number
   ```

   Le résultat d'une invite de commande Valkey ou Redis OSS ressemble à ce qui suit :

   ```
   primary-endpoint:port number
   ```

2. Vous pouvez désormais exécuter des commandes Valkey ou Redis OSS.

   ```
   set x Hello
   OK

   get x
   "Hello"
   ```

Connexion à un cluster en mode unencrypted-cluster activé

1. Exécutez la commande suivante pour vous connecter au cluster et remplacez *configuration-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey ou Redis OSS est 6379.)

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

> **Note**
>
> Dans la commande précédente, l'option -c active le mode cluster après les redirections -ASK et -MOVED.

Le résultat d'une invite de commande Valkey ou Redis OSS ressemble à ce qui suit :

```
configuration-endpoint:port number
```

2. Vous pouvez désormais exécuter des commandes Valkey ou Redis OSS. Notez que la redirection se produit car vous l'avez activée à l'aide de l'option -c. Si la redirection n'est pas activée, la commande renvoie l'erreur MOVED. Pour plus d'informations sur l'erreur MOVED, consultez la spécification du cluster Redis OSS.

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Connexion à un cluster Encryption/Authentication activé

Par défaut, valkey-cli utilise une connexion TCP non cryptée lors de la connexion à Valkey ou Redis OSS. L'option est `BUILD_TLS=yes` activée SSL/TLS au moment de la compilation de valkey-cli, comme indiqué dans la section précédente. Téléchargez et configurez l'accès en ligne de commande

L'activation d'AUTH est facultative. Toutefois, vous devez activer le chiffrement en transit pour activer AUTH. Pour plus de détails sur le ElastiCache chiffrement et l'authentification, consultezElastiCache chiffrement en transit (TLS).

> **Note**
>
> Vous pouvez utiliser l'option `--tls` avec valkey-cli pour vous connecter aux clusters chiffrés activés et désactivés en mode cluster. Si un cluster a un jeu de jetons AUTH, vous pouvez utiliser l'option `-a` pour fournir un mot de passe AUTH.

Dans les exemples suivants, veillez à remplacer *`cluster-endpoint`* et *`port number`* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey ou Redis OSS est 6379.)

Se connecter aux clusters chiffrés désactivés en mode cluster

L'exemple suivant se connecte à un cluster activé pour le chiffrement et l'authentification :

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

L'exemple suivant se connecte à un cluster dont le chiffrement est activé uniquement :

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Se connecter à des clusters chiffrés activés en mode cluster

L'exemple suivant se connecte à un cluster activé pour le chiffrement et l'authentification :

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

L'exemple suivant se connecte à un cluster dont le chiffrement est activé uniquement :

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Une fois connecté au cluster, vous pouvez exécuter les commandes Valkey ou Redis OSS comme indiqué dans les exemples précédents pour les clusters non chiffrés.

alternative à valkey-cli

Si le mode cluster n'est pas activé sur le cluster et que vous devez établir une connexion au cluster pour un court test, mais sans passer par la compilation valkey-cli, vous pouvez utiliser telnet ou openssl. Dans les exemples de commandes suivants, veillez à remplacer *cluster-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey ou Redis OSS est 6379.)

L'exemple suivant établit une connexion à un cluster dont l' and/or authentification par chiffrement est activée en mode cluster et dont le mode cluster est désactivé :

```
openssl s_client -connect cluster-endpoint:port number
```

Si un mot de passe est défini dans le cluster, connectez-vous d'abord au cluster. Après la connexion, authentifiez le cluster à l'aide de la commande suivante, puis appuyez sur la touche `Enter`. Dans l'exemple suivant, remplacez-le *your-password* par le mot de passe de votre cluster.

```
Auth your-password
```

L'exemple suivant se connecte à un cluster désactivé en mode cluster pour lequel le chiffrement ou l'authentification n'est pas activé :

```
telnet cluster-endpoint port number
```

Connectez-vous à un cluster ou à un groupe de réplication Valkey ou Redis OSS (Windows)

Pour vous connecter au cluster Valkey ou Redis OSS à partir d'une instance EC2 Windows à l'aide de la CLI Valkey ou de la CLI Redis OSS, vous devez télécharger le package valkey-cli et utiliser valkey-cli.exe pour vous connecter au cluster Valkey ou Redis OSS à partir d'une instance Windows. EC2

Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster dont le chiffrement n'est pas activé et qui exécute Valkey ou Redis OSS. Pour plus d'informations sur Valkey ou Redis OSS et les commandes disponibles, consultez les commandes Valkey et Redis OSS sur le site Web de Valkey.

Pour vous connecter à un cluster Valkey ou Redis OSS qui n'est pas activé pour le chiffrement à l'aide de valkey-cli

1. Connectez-vous à votre EC2 instance Amazon à l'aide de l'utilitaire de connexion de votre choix. Pour savoir comment vous connecter à une EC2 instance Amazon, consultez le guide de EC2 démarrage Amazon.
2. Copiez et collez le lien https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip dans un navigateur Internet pour télécharger le fichier zip du client Valkey à partir de la version disponible sur GitHub https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504

   Extrayez le fichier zip dans le dossier/chemin de votre choix.

   Ouvrez l'invite de commande, accédez au répertoire Valkey et exécutez la commandec`:\Valkey>valkey-cli -h` *`Redis_Cluster_Endpoint`* `-p 6379`.

   Par exemple :

   ```
   c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
   ```

3. Exécutez les commandes Valkey ou Redis OSS.

   Vous êtes maintenant connecté au cluster et pouvez exécuter les commandes Valkey ou Redis OSS comme suit.

   ```
   set a "hello"          // Set key "a" with a string value and no expiration
   OK
   get a                  // Get value for key "a"
   "hello"
   get b                  // Get value for key "b" results in miss
   (nil)
   set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
   "Good-bye"
   get b                  // Get value for key "b"
   "Good-bye"
                          // wait >= 5 seconds
   get b
   (nil)                  // key has expired, nothing returned
   quit                   // Exit from valkey-cli
   ```

Comment procéder ensuite ?

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

- Commencer avec AWS
- Outils pour Amazon Web Services
- Interface de ligne de commande AWS
- Référence ElastiCache d'API Amazon

Après avoir terminé l'exercice de mise en route, vous pouvez lire les sections suivantes pour en savoir plus sur ElastiCache l'administration :

- Choix de la taille de votre nœud

  Vous voulez que votre cache d'être suffisamment grand pour pouvoir accueillir toutes les données que vous souhaitez mettre en cache. En même temps, vous ne souhaitez pas payer pour plus d'espace de cache que ce dont vous avez besoin. Utilisez cette rubrique pour sélectionner la meilleure taille de nœud.
- ElastiCache meilleures pratiques et stratégies de mise en cache

  Identifier et résoudre les problèmes qui peuvent affecter l'efficacité de votre cluster.

Conception de votre propre cluster ElastiCache Redis OSS

Voici les actions ponctuelles que vous devez effectuer pour concevoir votre propre cluster ElastiCache Redis OSS.

Pour plus d'informations sur la configuration, ElastiCache voirCon ElastiCachefiguration.

Rubriques

- Étape 1 : Créer un groupe de sous-réseaux
- Étape 2 : Créer un cluster
- Étape 3 : Autoriser l'accès au cluster
- Étape 4 : Se connecter à un nœud de cluster

Étape 1 : Créer un groupe de sous-réseaux

Avant de créer votre cluster, créez tout d'abord un groupe de sous-réseaux. Un groupe de sous-réseaux de cache est un ensemble de sous-réseaux que vous souhaitez indiquer dans vos clusters de cache dans un VPC. Lors du lancement d'un cluster de cache dans un VPC, vous devez sélectionner un groupe de sous-réseaux de cache. ElastiCache Utilise ensuite ce groupe de sous-réseaux de cache pour attribuer des adresses IP au sein de ce sous-réseau à chaque nœud de cache du cluster.

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 expliquent comment créer un groupe de sous-réseaux appelé `mysubnetgroup` à l'aide de la console et de l' AWS CLI.

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 AWS de gestion et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste de navigation, sélectionnez Subnet Groups.
3. Choisissez Créer un groupe de sous-réseaux.
4. Dans l'assistant Create Subnet Group, procédez comme suit. Lorsque tous les paramètres sont définis selon vos choix, choisissez Yes, Create.

   a. Dans le champ Name, saisissez le nom de votre groupe de sous-réseaux de
   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.
   d. Dans les listes Availability Zone (Zone de disponibilité) et Subnet ID (ID de sous-réseau), choisissez la zone de disponibilité ou la zone locale et l'ID de votre sous-réseau privé, puis choisissez Add (Ajouter).

5. 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 ElastiCache console. 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.

Créer un groupe de sous-réseaux (AWS CLI)

A l'invite de commande, utilisez la commande `create-cache-subnet-group` pour créer un groupe de sous-réseaux de

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
```

```
    --subnet-ids subnet-53df9c3a
```

Pour Windows :

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubriquecreate-cache-subnet-group.

Étape 2 : 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 ElastiCache. Pour les besoins de cet exercice de mise en route, vous allez créer un cluster avec le mode cluster désactivé et 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). Les frais d' ElastiCache utilisation standard de l'instance vous seront facturés 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 ElastiCache d'utilisation, consultez Amazon ElastiCache.

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

Création d'un cluster Redis OSS (mode cluster désactivé) (console)

Pour créer un cluster Redis OSS (mode cluster désactivé) à l'aide de la console ElastiCache

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région dans laquelle vous souhaitez lancer ce cluster.
3. Choisissez Get started (Mise en route) dans le panneau de navigation.
4. Choisissez Create VPC (Créer un VPC) et suivez les étapes décrites à la page Création d'un cloud privé virtuel (VPC).
5. Sur la page du ElastiCache tableau de bord, choisissez Valkey cache ou Redis OSS cache. Pour cet exercice, nous allons choisir le cache Redis OSS, puis créer le cache Redis OSS.
6. Sous Cluster settings (Paramètres du cluster), procédez comme suit :
   a. Choisissez Configure and create a new cluster (Configurer et créer un cluster).
   b. Pour Cluster mode (Mode du cluster), choisissez Disabled (Désactivé).
   c. Pour Cluster info (Informations sur le cluster), saisissez une valeur pour Name (Nom).
   d. (Facultatif) Saisissez une valeur pour Description.
7. Sous Location (Emplacement) :

   AWS Cloud

   1. Pour AWS Cloud, nous vous recommandons d'accepter les paramètres par défaut pour Multi-AZ et Auto-failover (Basculement automatique). Pour plus d'informations, consultez Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ.
   2. Sous Cluster settings (Paramètres de cluster)
      a. Pour Engine version (Version du moteur), choisissez une version disponible.
      b. Pour Port, utilisez le port par défaut, 6379. Si vous avez une raison d'utiliser un autre port, saisissez le numéro de port.

c. Pour Groupe de paramètres, choisissez un groupe de paramètres ou créez-en un nouveau. Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres OSS Valkey et Redis et Création d'un groupe ElastiCache de paramètres.

> **Note**
>
> Lorsque vous sélectionnez un groupe de paramètres pour définir les valeurs de configuration du moteur, ce groupe de paramètres est appliqué à tous les clusters du magasin de données global. Dans la page Parameter Groups (Groupes de paramètres) l'attribut Global yes/no (oui/non) indique si un groupe de paramètres fait partie d'un magasin de données global.

d. Pour Type de nœud, choisissez la flèche vers le bas ( ). Dans la boîte de dialogue Modifier le type de nœud choisissez une valeur pour la famille d'instances pour le type de nœud souhaité. Choisissez ensuite le type de nœud que vous souhaitez utiliser pour ce cluster, puis choisissez Enregistrer.

   Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

   Si vous choisissez un type de nœud r6gd, la hiérarchisation des données est automatiquement activée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

e. Pour Number of replicas (Nombre de réplicas), choisissez le nombre de réplicas en lecture que vous voulez. Si vous avez activé Multi-AZ, le nombre doit être compris entre 1 et 5.

3. Sous Connectivity (Connectivité)

   a. Dans Network type (Type de réseau), choisissez la ou les versions d'adresses IP que ce cluster prendra en charge.

   b. Pour les groupes de sous-réseaux, choisissez le sous-réseau que vous souhaitez appliquer à ce cluster. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds. ElastiCache les clusters nécessitent un sous-réseau à double pile auquel les deux

IPv6 adresses sont attribuées pour fonctionner en mode double pile IPv4 et un sous-réseau « uniquement » pour fonctionner en tant que « IPv6 -only ». IPv6

Lorsque vous créez un nouveau groupe de sous-réseaux, entrez le VPC ID (ID du VPC) auquel il appartient.

Pour plus d'informations, consultez :

- Choix d'un type de réseau dans ElastiCache.
- Créer un sous-réseau dans votre VPC

Si vous utilisez Utilisation de zones locales avec ElastiCache, vous devez créer ou choisir un sous-réseau qui se trouve dans la zone locale.

Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux.

4. Pour Availability zone placements (Placement de zones de disponibilité), vous avez deux options :

   - Aucune préférence : ElastiCache choisit la zone de disponibilité.
   - Specify availability zones (Spécifier les zones de disponibilité) : vous spécifiez la zone de disponibilité pour chaque cluster.

     Si vous avez choisi de spécifier les Zones de disponibilité, pour chaque cluster de chaque partition, choisissez la Zone de disponibilité depuis la liste.

   Pour de plus amples informations, veuillez consulter Choix des régions et des zones de disponibilité pour ElastiCache.

5. Choisissez Next (Suivant)
6. Sous Paramètres avancés de Redis OSS

   - Pour Security (Sécurité) :

     i. 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 autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour de plus amples informations, veuillez consulter Utilisation clés AWS KMS gérées par le client.

- Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour la version 6.0 ElastiCache du moteur pour Redis OSS et les versions ultérieures, si vous activez le chiffrement en transit, vous serez invité à spécifier l'une des options de contrôle d'accès suivantes :
  - No Access Control (Aucun contrôle d'accès) – il s'agit du paramètre par défaut. Cela indique qu'aucune restriction n'est imposée à l'accès des utilisateurs au cluster.
  - User Group Access Control List (Liste de contrôle d'accès au groupe d'utilisateurs) : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs pouvant accéder au cluster. Pour de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.
  - Utilisateur par défaut AUTH — Mécanisme d'authentification pour les serveurs Valkey et Redis OSS. Pour plus d'informations, consultez AUTH.
- AUTH — Mécanisme d'authentification pour le serveur Redis OSS. Pour plus d'informations, consultez AUTH.

> **Note**
>
> Pour les versions de Redis OSS supérieures à 3.2.6, à l'exception de la version 3.2.10, Redis OSS AUTH est la seule option.

ii. 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.

7. Pour des sauvegardes automatiques régulières, choisissez Activer les sauvegardes automatiques, puis entrez le nombre de jours pendant lesquels vous souhaitez conserver une sauvegarde automatique avant sa suppression automatique. Si vous ne souhaitez pas de sauvegardes automatiques régulières, désactivez la case à cocher Enable automatic backups. Dans les deux cas, vous avez toujours la possibilité de créer des sauvegardes manuelles.

   Pour plus d'informations sur la sauvegarde et la restauration, consultezInstantané et restauration.

8. (Facultatif) Spécifiez une fenêtre de maintenance. La fenêtre de maintenance indique le moment, généralement d'une durée d'une heure, chaque semaine, où ElastiCache planifie la maintenance du système pour votre cluster. Vous pouvez autoriser ElastiCache à choisir le jour et l'heure de la fenêtre de maintenance (Aucune préférence) ou vous pouvez choisir vous-même le jour, l'heure et la durée (Specify maintenance window (Spécifier 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 de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

9. (Facultatif) Pour Logs (Journaux) :

   - Sous Log format (Format de journal), sélectionnez Text (Texte) ou JSON.
   - Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
   - Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatch journaux ou le nom de votre flux Firehose, ou choisissez Select existing, puis choisissez le nom de votre groupe de journaux de CloudWatch journaux ou le nom de votre stream Firehose,

10. Pour les balises, pour vous aider à gérer vos clusters et autres ElastiCache ressources, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Pour plus d'informations, consultez Marquer vos ressources ElastiCache .

11. Choisissez Suivant.

12. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create (Créer).

On premises

1. Pour On premises (Sur site), nous vous recommandons de laisser Auto-failover (Basculement automatique) activé. Pour plus d'informations, voir Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ
2. Pour terminer la création du cluster, suivez les étapes de la section Utilisation d'Outposts.

Dès que le statut de votre cluster est disponible, vous pouvez autoriser Amazon à y EC2 accéder, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 3. Autoriser l'accès au cluster et Étape 4 : Connectez-vous au nœud du cluster.

> **Important**
>
> Une fois que votre cluster est disponible, vous êtes facturé pour chaque heure ou partie d'heure 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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster Redis OSS (mode cluster désactivé) (AWS CLI)

Example

Le code CLI suivant crée un cluster de cache Redis OSS (mode cluster désactivé) sans répliques.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine redis \
--num-cache-nodes 1 \
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine redis ^
--num-cache-nodes 1 ^
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Pour utiliser le mode cluster activé, veuillez consulter les rubriques suivantes :

- Pour utiliser la console, veuillez consulter Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console).
- Pour utiliser le AWS CLI, voirCréation d'un cluster Valkey ou Redis OSS (mode cluster activé) ()AWS CLI.

Étape 3 : Autoriser l'accès au cluster

Cette section part du principe que vous êtes familiarisé avec le lancement et la connexion aux EC2 instances Amazon. Pour plus d'informations, consultez le guide de EC2 démarrage Amazon.

Tous les ElastiCache clusters sont conçus pour être accessibles depuis une EC2 instance Amazon. Le scénario le plus courant consiste à accéder à un ElastiCache cluster depuis une EC2 instance Amazon dans le même Amazon Virtual Private Cloud (Amazon VPC), ce qui sera le cas pour cet exercice.

Par défaut, l'accès réseau à votre cluster est limité au compte qui a été utilisé pour le créer. Avant de pouvoir vous connecter à un cluster à partir d'une EC2 instance, vous devez autoriser l' EC2 instance à accéder au cluster. Les étapes requises varient selon que vous avez lancé votre cluster dans EC2 -VPC ou EC2 -Classic.

Le cas d'utilisation le plus courant est celui où une application déployée sur une EC2 instance doit se connecter à un cluster dans le même VPC. La méthode la plus simple pour gérer l'accès entre les EC2 instances et les clusters d'un même VPC consiste à effectuer les opérations suivantes :

1. Créez un groupe de sécurité VPC pour votre cluster. Ce groupe de sécurité peut être utilisé pour restreindre l'accès aux instances de cluster. 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 Redis OSS et les groupes de réplication est`6379`.

   > **Important**
   >
   > Les groupes ElastiCache de sécurité Amazon ne s'appliquent qu'aux clusters qui ne s'exécutent pas dans un environnement Amazon Virtual Private Cloud (VPC). Si vous exécutez un Amazon Virtual Private Cloud, les groupes de sécurité ne sont pas disponibles dans le panneau de navigation de la console.
   > Si vous exécutez vos ElastiCache nœuds dans un Amazon VPC, vous contrôlez l'accès à vos clusters à l'aide des groupes de sécurité Amazon VPC, qui sont différents des groupes de sécurité. ElastiCache Pour plus d'informations sur l'utilisation ElastiCache dans un Amazon VPC, consultez Amazon VPCs et la ElastiCache sécurité

2. Créez un groupe de sécurité VPC pour vos EC2 instances (serveurs Web et d'applications). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l' EC2 instance depuis 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' EC2 instance via 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 EC2 instances. N'importe quel membre du groupe de sécurité peut ainsi accéder aux clusters.

> **Note**
>
> Si vous prévoyez de les utiliserUtilisation de zones locales avec ElastiCache, assurez-vous de les avoir activées. Lorsque vous créez un groupe de sous-réseaux dans cette zone locale, votre VPC est étendu à cette zone locale et votre VPC traitera le sous-réseau comme n'importe quel sous-réseau dans n'importe quelle autre zone de disponibilité. Toutes les passerelles et les tables de routage pertinentes seront automatiquement ajustées.

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 https://console.aws.amazon.com l'adresse /vpc.
2. Dans le panneau de navigation, choisissez Groupes de sécurité.
3. Sélectionnez ou créez un groupe de sécurité que vous utiliserez pour vos instances cluster. 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 Redis OSS et les groupes de réplication est6379.

   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 EC2 instances Amazon.
5. Choisissez Enregistrer lorsque vous avez terminé.

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

Pour plus d'informations sur l'accès à votre ElastiCache cluster 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 ElastiCache cache dans un Amazon VPC
- Accès aux ElastiCache ressources depuis l'extérieur AWS

Étape 4 : Se connecter à un nœud de cluster

Avant de continuer, terminez la section Étape 3 : Autoriser l'accès au cluster.

Cette section part du principe que vous avez créé une EC2 instance Amazon et que vous pouvez vous y connecter. Pour obtenir des instructions sur la procédure à suivre, consultez le guide de EC2 démarrage Amazon.

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

Rechercher vos points de terminaison de nœud

Lorsque votre cluster est disponible et que vous y avez un accès autorisé, vous pouvez vous connecter à une EC2 instance Amazon et vous connecter au cluster. Pour cela, vous devez d'abord déterminer le point de terminaison.

Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)

Si un cluster Redis OSS (mode cluster désactivé) ne possède qu'un seul nœud, le point de terminaison du nœud est utilisé à la fois pour les lectures et les écritures. Si un cluster a plusieurs nœuds, il y a trois types de points de terminaison : le point de terminaison principal, le point de terminaison du lecteur et les points de terminaison des nœuds.

Le point de terminaison principal est un nom DNS qui est toujours résolu dans le nœud principal du cluster. Le point de terminaison principal n'est pas affecté par les changements apportés à votre cluster, tels que la promotion d'un réplica en lecture au rang de réplica principal. Pour l'activité d'écriture, il est préférable que vos applications se connectent au point de terminaison principal.

Un point de terminaison de lecteur répartira de manière égale les connexions entrantes vers le point de terminaison entre toutes les répliques de lecture ElastiCache d'un cluster Redis OSS. Des facteurs supplémentaires, comme le moment où l'application crée les connexions ou la façon dont l'application utilise ou réutilise les connexions, détermineront la distribution du trafic. Les points de terminaison du lecteur suivent les changements de cluster en temps réel au fur et à mesure que des réplicas sont ajoutés ou supprimés. Vous pouvez placer ElastiCache les multiples répliques de lecture de votre cluster Redis OSS dans différentes zones de AWS disponibilité (AZ) afin de garantir la haute disponibilité des points de terminaison des lecteurs.

> **Note**
>
> Un point de terminaison du lecteur n'est pas un équilibreur de charge. Il s'agit d'un enregistrement DNS qui se résoudra à une adresse IP de l'un des nœuds de réplica en mode round robin.

Pour l'activité de lecture, les applications peuvent également se connecter à n'importe quel nœud du cluster. Contrairement au point de terminaison principal, les points de terminaison des nœuds sont résolus dans des points de terminaison spécifiques. En cas de modification dans votre cluster, comme l'ajout ou la suppression d'un réplica, vous devez mettre à jour les points de terminaison des nœuds de votre application.

Pour rechercher les points de terminaison d'un cluster Redis OSS (mode cluster désactivé)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez les caches Redis OSS.

   L'écran des clusters apparaîtra avec une liste qui inclura tous les caches sans serveur Valkey ou Redis OSS existants, les clusters Redis OSS (mode cluster désactivé) et les clusters Redis OSS (mode cluster activé). Choisissez le cluster que vous avez créé dans la section Création d'un cluster Redis OSS (mode cluster désactivé) (console).
3. Pour trouver les points de terminaison du and/or lecteur principal du cluster, choisissez le nom du cluster (et non le bouton radio).

Points de terminaison principaux et lecteurs pour un cluster Redis OSS (mode cluster désactivé)

S'il y a un seul nœud dans le cluster, il n'y a aucun point de terminaison principal et vous pouvez passer à l'étape suivante.

4. Si le cluster Redis OSS (mode cluster désactivé) possède des nœuds de réplication, vous pouvez trouver les points de terminaison des nœuds de réplication du cluster en choisissant le nom du cluster, puis en choisissant l'onglet Nœuds.

   L'écran des nœuds s'affiche avec chaque nœud du cluster, principal et de réplica, répertorié avec son point de terminaison.

Points de terminaison de nœuds pour un cluster Redis OSS (mode cluster désactivé)

5. Pour copier un point de terminaison dans le presse-papiers :

   a. Un point de terminaison à la fois, recherchez le point de terminaison que vous souhaitez copier.

   b. Cliquez sur l'icône de copie située juste devant le point de terminaison.

Le point de terminaison est maintenant copié dans votre presse-papiers. Pour en savoir plus sur l'utilisation du point de terminaison pour se connecter à un nœud, consultez Connexion aux nœuds.

Un point de terminaison principal Redis OSS (mode cluster désactivé) ressemble à ce qui suit. Il existe une différence selon que le chiffrement en transit est activé ou pas.

Chiffrement en transit non activé

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Chiffrement en transit activé

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Pour en savoir plus sur la recherche de vos points de terminaison, consultez les rubriques correspondant au moteur et au type de cluster que vous exécutez.

- Recherche de points de terminaison de connexion dans ElastiCache
- Recherche de points de terminaison pour un cluster Valkey ou Redis OSS (mode cluster activé) (console) – vous avez besoin du point de terminaison de la configuration du cluster.
- Recherche des points de terminaison (AWS CLI)
- Recherche de points de terminaison (ElastiCache API)

Connectez-vous à un cluster ou à un groupe de réplication Valkey ou Redis OSS (Linux)

Maintenant que vous disposez du point de terminaison dont vous avez besoin, vous pouvez vous connecter à une EC2 instance et vous connecter au cluster ou au groupe de réplication. Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster. La dernière version de valkey-cli permet également de connecter SSL/TLS encryption/authentication des clusters activés.

L'exemple suivant utilise des EC2 instances Amazon exécutant Amazon Linux et Amazon Linux 2. Pour plus de détails sur l'installation et la compilation de valkey-cli avec d'autres distributions Linux, consultez la documentation de votre système d'exploitation spécifique.

> **Note**
>
> Ce processus couvre le test d'une connexion à l'aide de l'utilitaire valkey-cli pour une utilisation imprévue uniquement. Pour obtenir la liste des clients pris en charge, consultez la documentation de Valkey. Pour des exemples d'utilisation du AWS SDKs with ElastiCache, voirTutoriels : Débuter avec Python et ElastiCache.

Connexion à un mode cluster désactivé cluster non chiffré

1. Exécutez la commande suivante pour vous connecter au cluster et remplacez *primary-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey et Redis OSS est 6379.)

   ```
   src/valkey-cli -h primary-endpoint -p port number
   ```

   Le résultat affiché dans une invite de commande ressemble à ce qui suit :

   ```
   primary-endpoint:port number
   ```

2. Vous pouvez désormais exécuter les commandes Valkey et Redis OSS.

   ```
   set x Hello
   OK

   get x
   "Hello"
   ```

Connexion à un cluster en mode unencrypted-cluster activé

1. Exécutez la commande suivante pour vous connecter au cluster et remplacez *configuration-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Valkey et Redis OSS est 6379.)

```
src/valkey-cli -h configuration-endpoint -c -p port number
```

> **Note**
>
> Dans la commande précédente, l'option -c active le mode cluster après les redirections -ASK et -MOVED.

Le résultat affiché dans une invite de commande ressemble à ce qui suit :

```
configuration-endpoint:port number
```

2. Vous pouvez désormais exécuter les commandes Valkey et Redis OSS. Notez que la redirection se produit car vous l'avez activée à l'aide de l'option -c. Si la redirection n'est pas activée, la commande renvoie l'erreur MOVED. Pour plus d'informations sur l'erreur MOVED, consultez la spécification du cluster.

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
"Hi"
```

Connexion à un cluster Encryption/Authentication activé

Par défaut, valkey-cli utilise une connexion TCP non cryptée lors de la connexion à Valkey et Redis OSS. L'option est `BUILD_TLS=yes` activée SSL/TLS au moment de la compilation de valkey-cli, comme indiqué dans la section précédente. Téléchargez et configurez l'accès en ligne de commande

L'activation d'AUTH est facultative. Toutefois, vous devez activer le chiffrement en transit pour activer AUTH. Pour plus de détails sur le ElastiCache chiffrement et l'authentification, consultezElastiCache chiffrement en transit (TLS).

> **Note**
>
> Vous pouvez utiliser l'option `--tls` avec valkey-cli pour vous connecter aux clusters chiffrés activés et désactivés en mode cluster. Si un cluster a un jeu de jetons AUTH, vous pouvez utiliser l'option `-a` pour fournir un mot de passe AUTH.

Dans les exemples suivants, veillez à remplacer *`cluster-endpoint`* et *`port number`* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Redis OSS est 6379.)

Se connecter aux clusters chiffrés désactivés en mode cluster

L'exemple suivant se connecte à un cluster activé pour le chiffrement et l'authentification :

```
src/valkey-cli -h cluster-endpoint --tls -a your-password -p port number
```

L'exemple suivant se connecte à un cluster dont le chiffrement est activé uniquement :

```
src/valkey-cli -h cluster-endpoint --tls -p port number
```

Se connecter à des clusters chiffrés activés en mode cluster

L'exemple suivant se connecte à un cluster activé pour le chiffrement et l'authentification :

```
src/valkey-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

L'exemple suivant se connecte à un cluster dont le chiffrement est activé uniquement :

```
src/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Une fois connecté au cluster, vous pouvez exécuter les commandes Valkey ou Redis OSS comme indiqué dans les exemples précédents pour les clusters non chiffrés.

Une alternative à valkey-cli ou à Redis-cli

Si le mode cluster n'est pas activé sur le cluster et que vous devez établir une connexion au cluster pour un court test, mais sans passer par la compilation valkey-cli ou redis-cli, vous pouvez utiliser telnet ou openssl. Dans les exemples de commandes suivants, veillez à remplacer *cluster-endpoint* et *port number* par le point de terminaison de votre cluster et votre numéro de port. (Le port par défaut pour Redis OSS est 6379.)

L'exemple suivant établit une connexion à un cluster dont l' and/or authentification par chiffrement est activée en mode cluster et dont le mode cluster est désactivé :

```
openssl s_client -connect cluster-endpoint:port number
```

Si un mot de passe est défini dans le cluster, connectez-vous d'abord au cluster. Après la connexion, authentifiez le cluster à l'aide de la commande suivante, puis appuyez sur la touche `Enter`. Dans l'exemple suivant, remplacez-le *your-password* par le mot de passe de votre cluster.

```
Auth your-password
```

L'exemple suivant se connecte à un cluster désactivé en mode cluster pour lequel le chiffrement ou l'authentification n'est pas activé :

```
telnet cluster-endpoint port number
```

Connectez-vous à un cluster ou à un groupe de réplication Valkey ou Redis OSS (Windows)

Pour vous connecter au cluster depuis une instance EC2 Windows à l'aide de la CLI Valkey ou Redis OSS, vous devez télécharger le package valkey-cli et utiliser valkey-cli.exe pour vous connecter au cluster Valkey ou Redis OSS à partir d'une instance Windows. EC2

Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster dont le chiffrement n'est pas activé et qui exécute Valkey ou Redis OSS. Pour plus d'informations sur Valkey et les commandes disponibles, consultez la section Commandes Valkey sur le site Web de Valkey.

Pour vous connecter à un cluster Valkey ou Redis OSS qui n'est pas activé pour le chiffrement à l'aide de valkey-cli

1. Connectez-vous à votre EC2 instance Amazon à l'aide de l'utilitaire de connexion de votre choix. Pour obtenir des instructions sur la façon de se connecter à une EC2 instance Amazon, consultez le guide de EC2 démarrage Amazon.

2. Copiez et collez le lien https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip dans un navigateur Internet pour télécharger le fichier zip du client Redis OSS à partir de la version disponible sur GitHub https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504

   Extrayez le fichier zip dans le dossier/chemin de votre choix.

   Ouvrez l'invite de commande, accédez au répertoire Valkey et exécutez la commande`c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379`.

   Par exemple :

```
c:\Valkey>valkey-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. Exécutez les commandes Valkey ou Redis OSS.

   Vous êtes maintenant connecté au cluster et pouvez exécuter les commandes Valkey ou Redis OSS comme suit.

```
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
get b                  // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                  // Get value for key "b"
"Good-bye"
                       // wait >= 5 seconds
get b
(nil)                  // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Suppression d'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 ElastiCache cluster, vos instantanés manuels sont conservés. Vous pouvez également créer un instantané final avant la suppression du cluster. Les instantanés mis en cache automatiquement ne sont pas conservés. Pour de plus amples informations, veuillez consulter Instantané et restauration.
> - `CreateSnapshot`une autorisation est requise pour créer un instantané final. Sans cette autorisation, l'appel d'API échouera avec une `Access Denied` exception.

En utilisant le 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 ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le tableau de bord ElastiCache du moteur, choisissez Valkey ou Redis OSS.

   La liste de tous les caches exécutés sur ce moteur apparaît.
3. Pour choisir le cluster à supprimer, choisissez le nom du cluster dans la liste des clusters. Dans ce cas, le nom du cluster que vous avez créé sur Étape 2 : Créer un cluster.

   > **Important**
   >
   > Vous ne pouvez supprimer qu'un seul cluster à la fois depuis la ElastiCache console. Le choix de plusieurs clusters désactive l'opération de suppression.
4. Pour Actions, choisissez Supprimer.
5. Dans l'écran de confirmation Supprimer un cluster, entrez le nom du cluster et choisissez Sauvegarde finale. Choisissez ensuite 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.

En utilisant le AWS CLI

Le code suivant supprime le cluster de cache `my-cluster`. Dans ce cas, remplacez `my-cluster` par le nom du cluster que vous avez créé sur Étape 2 : Créer un cluster.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

L'action `delete-cache-cluster` de la CLI supprime uniquement un cluster de cache. Pour supprimer plusieurs clusters de cache, appelez `delete-cache-cluster` pour chaque cluster de cache devant être supprimé. Vous n'avez pas besoin d'attendre que la suppression d'un cluster de cache soit terminée avant d'en supprimer un autre.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id my-cluster \
    --region us-east-2
```

Pour Windows :

```
aws elasticache delete-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --region us-east-2
```

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée `delete-cache-cluster`.

Autres ElastiCache tutoriels et vidéos

Les didacticiels suivants abordent les tâches présentant un intérêt pour les ElastiCache utilisateurs d'Amazon.

- ElastiCache Vidéos
- Tutoriel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un Amazon VPC

ElastiCache Vidéos

Voici des vidéos qui vous aideront à apprendre les ElastiCache concepts de base et avancés d'Amazon. Pour plus d'informations sur AWS la formation, voir AWS Formation et certification.

Rubriques

- Vidéos d'introduction
- Vidéos avancées

Vidéos d'introduction

Les vidéos suivantes vous présentent Amazon ElastiCache.

Rubriques

- AWS re:Invent 2020 : les nouveautés d'Amazon ElastiCache
- AWS re:Invent 2019 : les nouveautés d'Amazon ElastiCache
- AWS re:Invent 2017 : les nouveautés d'Amazon ElastiCache
- DAT204 — Création d'applications évolutives sur les services AWS NoSQL (re:Invent 2015)
- DAT207 — Accélérer les performances des applications avec Amazon ElastiCache (AWS re:Invent 2013)

AWS re:Invent 2020 : les nouveautés d'Amazon ElastiCache

AWS re:Invent 2020 : les nouveautés d'Amazon ElastiCache

AWS re:Invent 2019 : les nouveautés d'Amazon ElastiCache

AWS re:Invent 2019 : les nouveautés d'Amazon ElastiCache

AWS re:Invent 2017 : les nouveautés d'Amazon ElastiCache

AWS re:Invent 2017 : les nouveautés d'Amazon ElastiCache

DAT204 — Création d'applications évolutives sur les services AWS NoSQL (re:Invent 2015)

Au cours de cette session, nous aborderons les avantages des bases de données NoSQL et passerons en revue les principaux services NoSQL proposés par AWS Amazon DynamoDB et Amazon. ElastiCache Ensuite, nous entendons deux clients de premier plan, Expedia et Mapbox,

nous parler de leurs cas d'utilisation et de leurs défis architecturaux, et de la manière dont ils les ont abordés à l'aide des services AWS NoSQL, notamment en ce qui concerne les modèles de conception et les meilleures pratiques. A la fin de cette session vous devriez avoir une meilleure compréhension de NoSQL et de ses capacités performantes, pour affronter avec confiance vos défis de gestion de bases de données.

DAT204 — Création d'applications évolutives sur les services AWS NoSQL (re:Invent 2015)

DAT207 — Accélérer les performances des applications avec Amazon ElastiCache (AWS re:Invent 2013)

Dans cette vidéo, découvrez comment utiliser Amazon ElastiCache pour déployer facilement un système de mise en cache en mémoire afin d'accélérer les performances de vos applications. Nous vous montrons comment utiliser Amazon ElastiCache pour améliorer la latence de vos applications et réduire la charge sur vos serveurs de base de données. Nous vous montrerons également comment créer une couche de mise en cache facile à gérer et à adapter au fur et à mesure que votre application se développe. Au cours de cette session, nous passerons en revue différents scénarios et cas d'utilisation qui peuvent bénéficier de l'activation de la mise en cache, et discutons des fonctionnalités proposées par Amazon ElastiCache.

DAT207 - Accélérer les performances des applications avec Amazon ElastiCache (re:Invent 2013)

Vidéos avancées

Les vidéos suivantes abordent des ElastiCache sujets Amazon plus avancés.

Rubriques

- Concevez pour réussir grâce aux ElastiCache meilleures pratiques d'Amazon (re:Invent 2020)
- Boostez vos applications en temps réel avec Amazon ElastiCache (re:Invent 2019)
- Bonnes pratiques : migration de clusters Redis OSS d'Amazon EC2 vers ElastiCache (re:Invent 2019)
- Développez une plateforme de sports fantastiques avec Amazon ElastiCache et Amazon Aurora STP11 (re:Invent 2018)
- Redis OSS fiable et évolutif dans le cloud avec Amazon ElastiCache (re:Invent 2018)
- ElastiCache Deep Dive : modèles de conception pour les stockages de données en mémoire (re:Invent 2018)
- DAT305 — Amazon ElastiCache Deep Dive (re:Invent 2017)
- DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

- DAT317—Comment IFTTT utilise Redis OSS ElastiCache pour prédire les événements (re:Invent 2016)
- DAT407 — Amazon ElastiCache Deep Dive (re:Invent 2015)
- SDD402 — Amazon ElastiCache Deep Dive (re:Invent 2014)
- DAT307 — Présentation approfondie de l' ElastiCache architecture et des modèles de conception d'Amazon (re:Invent 2013)

Concevez pour réussir grâce aux ElastiCache meilleures pratiques d'Amazon (re:Invent 2020)

Avec la croissance exponentielle des applications critiques en temps réel basées sur Redis OSS, la disponibilité, l'évolutivité et la sécurité sont devenues des considérations majeures. Découvrez les meilleures pratiques pour garantir le succès d'Amazon grâce à ElastiCache la mise à l'échelle en ligne, à la haute disponibilité pour les déploiements multi-AZ et aux configurations de sécurité.

Concevez pour réussir grâce aux ElastiCache meilleures pratiques d'Amazon (re:Invent 2020)

Boostez vos applications en temps réel avec Amazon ElastiCache (re:Invent 2019)

Avec la croissance rapide de l'adoption du cloud et les nouveaux scénarios qu'elles autorisent, les applications ont besoin d'une latence de l'ordre de la microseconde et d'un débit élevé pour prendre en charge des millions de requêtes par seconde. Les développeurs se sont généralement appuyés sur du matériel spécialisé et des solutions de contournement, telles que des bases de données sur disque combinées à des techniques de réduction des données, pour gérer les données des applications en temps réel. Ces approches peuvent être coûteuses et non évolutives. Découvrez comment améliorer les performances des applications en temps réel en utilisant Amazon en mémoire entièrement géré ElastiCache pour des performances, une évolutivité, une disponibilité et une sécurité extrêmes.

Boostez vos applications en temps réel avec Amazon ElastiCache (re:Invent 2019:)

Bonnes pratiques : migration de clusters Redis OSS d'Amazon EC2 vers ElastiCache (re:Invent 2019)

Gérer soi-même les clusters Redis OSS peut s'avérer difficile. Vous devez approvisionner le matériel, corriger les logiciels, sauvegarder les données et surveiller les charges de travail en permanence. Grâce à la nouvelle fonctionnalité de migration en ligne pour Amazon ElastiCache, vous pouvez désormais facilement déplacer vos données depuis Redis OSS auto-hébergé sur Amazon EC2 vers Amazon entièrement géré ElastiCache, le mode cluster étant désactivé. Au cours de cette session, vous découvrirez le nouvel outil de migration en ligne, assisterez à une démonstration et,

plus important encore, découvrirez les meilleures pratiques pour une migration fluide vers Amazon ElastiCache.

Bonnes pratiques : migration de clusters Redis OSS d'Amazon EC2 vers ElastiCache (re:Invent 2019)

Développez une plateforme de sports fantastiques avec Amazon ElastiCache et Amazon Aurora STP11 (re:Invent 2018)

Dream11 est la première startup de technologie sportive en Inde. Il compte une base croissante de plus de 40 millions d'utilisateurs pratiquant de multiples sports, y compris le cricket fantastique, le football et le basket-ball, et il dessert actuellement un million d'utilisateurs simultanés, qui produisent trois millions de demandes par minute sous un temps de réponse de 50 millisecondes. Dans cette conférence, Amit Sharma, directeur technique de Dream11, explique comment l'entreprise utilise Amazon Aurora et Amazon ElastiCache pour gérer le trafic flash, qui peut tripler en 30 secondes. Sharma parle également de la mise à l'échelle des transactions sans verrouillage et partage les étapes de gestion du trafic Flash, desservant ainsi cinq millions d'utilisateurs actifs quotidiens. Titre complet : AWS re:Invent 2018 : Scaling a fantasy sports platform with Amazon ElastiCache & Amazon Aurora () STP11

Développez une plateforme de sports fantastiques avec Amazon ElastiCache et Amazon Aurora STP11 (re:Invent 2018)

Redis OSS fiable et évolutif dans le cloud avec Amazon ElastiCache (re:Invent 2018)

Cette session couvre les fonctionnalités et les améliorations de notre service compatible Redis OSS, Amazon ElastiCache for Redis OSS. Nous abordons les fonctionnalités clés, telles que Redis OSS 5, l'amélioration de l'évolutivité et des performances, la sécurité et la conformité, et bien plus encore. Nous abordons également les fonctions à venir, ainsi que des études de cas clients.

Redis OSS fiable et évolutif dans le cloud avec Amazon ElastiCache (re:Invent 2018)

ElastiCache Deep Dive : modèles de conception pour les stockages de données en mémoire (re:Invent 2018)

Au cours de cette session, nous vous donnons un aperçu des coulisses pour en savoir plus sur le design et l'architecture d'Amazon ElastiCache. Découvrez les modèles de conception courants de nos offres Redis OSS et Memcached et comment les clients les utilisent pour le traitement des données en mémoire afin de réduire la latence et d'améliorer le débit des applications. Nous examinons ElastiCache les meilleures pratiques, les modèles de conception et les anti-modèles.

ElastiCache Deep Dive : modèles de conception pour les stockages de données en mémoire (re:Invent 2018)

DAT305 — Amazon ElastiCache Deep Dive (re:Invent 2017)

Découvrez les coulisses pour en savoir plus sur le design et ElastiCache l'architecture d'Amazon. Découvrez les modèles de conception courants de nos offres Memcached et Redis OSS et comment les clients les ont utilisés pour les opérations en mémoire afin de réduire la latence et d'améliorer le débit des applications. Dans cette vidéo, nous passons en revue les ElastiCache meilleures pratiques, les modèles de conception et les anti-modèles.

La vidéo présente les fonctionnalités suivantes°:

- ElastiCache pour le repartage en ligne de Redis OSS
- ElastiCache sécurité et cryptage
- ElastiCache pour Redis OSS version 3.2.10

DAT305 — Amazon ElastiCache Deep Dive (re:Invent 2017)

DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

Découvrez les coulisses pour en savoir plus sur le design et ElastiCache l'architecture d'Amazon. Découvrez les modèles de conception courants de nos offres Memcached et Redis OSS et comment les clients les ont utilisés pour les opérations en mémoire afin de réduire la latence et d'améliorer le débit des applications. Au cours de cette session, nous passerons en revue les ElastiCache meilleures pratiques, les modèles de conception et les anti-modèles.

DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

DAT317—Comment IFTTT utilise Redis OSS ElastiCache pour prédire les événements (re:Invent 2016)

IFTTT est un service gratuit qui permet aux personnes d'effectuer davantage d'opérations avec les services qu'elles apprécient, de l'automatisation des tâches simples à la transformation de l'interaction d'une personne avec sa maison et de son contrôle sur celle-ci. IFTTT utilise Redis OSS ElastiCache pour stocker l'historique des transactions et les prévisions de planification, ainsi que les index des documents de journal sur Amazon S3. Visionnez cette session pour découvrir comment la puissance de script de Lua et les types de données de Redis OSS ont permis aux utilisateurs d'accomplir quelque chose qu'ils n'auraient pas pu faire ailleurs.

DAT317—Comment IFTTT utilise Redis OSS ElastiCache pour prédire les événements (re:Invent 2016)

DAT407 — Amazon ElastiCache Deep Dive (re:Invent 2015)

Jetez un coup d'œil dans les coulisses pour en savoir plus sur ElastiCache le design et l'architecture d'Amazon. Découvrez les modèles de conception courants de nos offres Memcached et Redis OSS et comment les clients les ont utilisés pour les opérations en mémoire et ont amélioré la latence et le débit des applications. Au cours de cette session, nous passerons en revue les meilleures pratiques, les modèles de conception et les anti-modèles liés à Amazon ElastiCache.

DAT407 — Amazon ElastiCache Deep Dive (re:Invent 2015)

SDD402 — Amazon ElastiCache Deep Dive (re:Invent 2014)

Dans cette vidéo, nous examinons les cas d'utilisation courants de la mise en cache, ElastiCache les moteurs Memcached et ElastiCache Redis OSS, les modèles qui vous aident à déterminer quel moteur répond le mieux à vos besoins, le hachage cohérent, etc. afin de créer des applications rapides et évolutives. Frank Wiebe, scientifique principal chez Adobe, explique comment Adobe utilise Amazon ElastiCache pour améliorer l'expérience client et développer son activité.

DAT402 — Amazon ElastiCache Deep Dive (re:Invent 2014)

DAT307 — Présentation approfondie de l' ElastiCache architecture et des modèles de conception d'Amazon (re:Invent 2013)

Cette vidéo aborde la mise en cache, les stratégies de mise en cache, l'évolution et la surveillance. Nous comparons également ElastiCache les moteurs Memcached et Redis ElastiCache OSS. Au cours de cette session, nous passerons également en revue les meilleures pratiques et les modèles de conception liés à Amazon ElastiCache.

DAT307 - Découvrez en profondeur l' ElastiCache architecture et les modèles de design d'Amazon (AWS re:Invent 2013).

Gestion des nœuds dans ElastiCache

Un nœud est le plus petit élément constitutif d'un ElastiCache déploiement Amazon. C'est une partie à taille fixe de RAM sécurisée attachée au réseau. Chaque nœud exécute le moteur qui a été choisi lors de la création du cluster ou de sa dernière modification. Chaque nœud a son propre port et nom DNS (Domain Name Service). Plusieurs types de ElastiCache nœuds sont pris en charge, chacun étant associé à des quantités variables de mémoire et de puissance de calcul.

Pour une discussion plus détaillée sur la taille de nœud à utiliser, consultez Choix de la taille de votre nœud.

D'une manière générale, en raison de sa prise en charge du sharding, les déploiements de Valkey ou Redis OSS (mode cluster activé) comportent un certain nombre de nœuds plus petits. En revanche, les déploiements Valkey ou Redis OSS (mode cluster désactivé) comportent moins de nœuds plus volumineux dans un cluster. Pour une discussion plus détaillée sur la taille de nœud à utiliser, consultez Choix de la taille de votre nœud.

Rubriques

- Affichage de l'état du ElastiCache nœud
- Nœuds et partitions Valkey ou Redis OSS
- Connexion aux nœuds
- Types de nœuds pris en charge
- Redémarrage des nœuds
- Remplacement de nœuds (Valkey et Redis OSS)
- Remplacement de nœuds (Memcached)
- Nœuds réservés
- Migration des nœuds de génération précédente

Voici certaines opérations importantes impliquant des nœuds :

- Ajouter des nœuds à un ElastiCache cluster
- Supprimer des nœuds d'un ElastiCache cluster
- Dimensionnement ElastiCache
- Recherche de points de terminaison de connexion dans ElastiCache
- Identifiez automatiquement les nœuds de votre cluster (Memcached)

Affichage de l'état du ElastiCache nœud

À l'aide de la ElastiCache console, vous pouvez accéder rapidement à l'état de votre ElastiCache nœud. L'état d'un ElastiCache nœud indique son état de santé. Vous pouvez utiliser les procédures suivantes pour consulter l'état du ElastiCache nœud dans la ElastiCache console Amazon, la AWS CLI commande ou le fonctionnement de l'API.

Les valeurs d'état possibles pour ElastiCache les nœuds figurent dans le tableau suivant. Ce tableau indique également si le ElastiCache nœud vous sera facturé.

Type	Facturé	Description
`available`	Facturé	Le ElastiCache nœud est sain et disponible.
`creating`	Non facturé	Le ElastiCache nœud est en cours de création. Le nœud n'est pas accessible pendant sa création.
`deleting`	Non facturé	Le ElastiCache nœud est en cours de suppression.
`modifying`	Facturé	Le ElastiCache nœud est en cours de modification suite à la demande d'un client.
`updating`	Facturé	Un état de mise à jour indique qu'une ou plusieurs des conditions suivantes sont vraies pour le ElastiCache nœud Amazon : • Le ElastiCache nœud est corrigé dans le cadre de la mise à jour du service. Pour plus d'informations sur les mises à jour de service, consultez la page d'aide Amazon ElastiCache Managed Maintenance and Service Updates. • Les groupes de sécurité VPC sont en cours de mise

Type	Facturé	Description
		à jour pour le ElastiCache cluster. • Le ElastiCache cluster est en cours d'agrandissement ou de réduction. • Les configurations de livraison des journaux sont en cours de modification pour le ElastiCache cluster. • Une opération de suppressi on du ElastiCache nœud est en attente. • Le ElastiCache mot de passe est en cours d' updated/rotated utilisation AWS Secrets Manager.
`rebooting cache cluster nodes`	Facturé	Le ElastiCache nœud est redémarré à la suite d'une demande d'un client ou d'un ElastiCache processus Amazon nécessitant le redémarrage du nœud.

Type	Facturé	Description
`incompatible_parameters`	Non facturé	Amazon ne ElastiCache peut pas démarrer le nœud car les paramètres spécifiés dans le groupe de paramètres du nœud ne sont pas compatibles avec le nœud. Annulez les modifications des paramètres ou rendez-les compatibles avec le nœud pour rétablir l'accès à votre nœud. Pour plus d'informations sur les paramètres incompatibles, consultez la liste des événements du ElastiCache nœud.
`incompatible_network`	Non facturé	Un état de réseau incompatible indique qu'une ou plusieurs des conditions suivantes sont vraies pour le nœud Amazon : ElastiCache • Aucune adresse IP n'est disponible dans le sous-réseau dans lequel le ElastiCache nœud a été lancé. • Le sous-réseau mentionné dans le groupe de ElastiCache sous-réseaux n'existe plus dans Amazon Virtual Private Cloud (Amazon VPC).

Type	Facturé	Description
`restore_failed`	Non facturé	Un état d'échec de restaurat ion indique que l'une des conditions suivantes est vraie pour le nœud Amazon : ElastiCache • Les remplacements de nœuds ont échoué à plusieurs reprises en raison d'une capacité d'instanc e insuffisante. Cela se produit généralement lors de l'exécution de nœuds de génération précédent e qui le sont end-of-li fe. Toutefois, cela peut également se produire lors du remplacement de nœuds de génération actuelle lorsque la capacité à la demande AWS n'est pas suffisante pour répondre à votre demande dans la zone de disponibilité spécifiée. Pour plus d'informations sur la réparation ou la suppressi on de ces nœuds, consultez Migration des nœuds de génération précédente. • La restauration de l'instant ané RDB spécifié a échoué. • Le AWS compte du ElastiCache cluster a été suspendu.

Type	Facturé	Description
		• Le nœud a échoué et n'a pas pu être restauré.
`snapshotting`	Facturé	ElastiCache crée un instantané du nœud Valkey ou Redis OSS.

Affichage de l'état du ElastiCache nœud avec la console

Pour afficher l'état d'un ElastiCache nœud avec la console :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Redis OSS Clusters ou Memcached Clusters. La page Caches apparaît avec la liste des ElastiCache nœuds. La valeur de statut est affichée pour chaque nœud.
3. Vous pouvez ensuite accéder à l'onglet Mises à jour du service pour le cache afin d'afficher la liste des mises à jour du service applicables au cache.

Affichage de l'état du ElastiCache nœud à l'aide du AWS CLI

Pour afficher le ElastiCache nœud et ses informations d'état à l'aide de AWS CLI, utilisez la `describe-cache-cluster` commande. Par exemple, la AWS CLI commande suivante affiche chaque ElastiCache nœud.

```
aws elasticache describe-cache-clusters
```

Affichage de l'état du ElastiCache nœud via l'API

Pour consulter l'état du ElastiCache nœud à l'aide de l' ElastiCache API Amazon, appelez l'`ShowCacheNodeInfo`indicateur `DescribeCacheClusteroperation` with the pour récupérer des informations sur les nœuds de cache individuels.

Nœuds et partitions Valkey ou Redis OSS

Une partition (qui équivaut à un groupe de nœuds dans l'API et dans la CLI) est une organisation hiérarchique des nœuds, chacune incluse dans un cluster. Les partitions prennent en charge la réplication. Au sein d'une partition, un nœud fonctionne comme nœud read/write principal. Tous les autres nœuds contenus dans une partition fonctionnent comme des réplicas en lecture seule du nœud primaire. Valkey, ou Redis OSS version 3.2 ou ultérieure, prend en charge plusieurs partitions au sein d'un cluster (dans l'API et la CLI, un groupe de réplication). Cette prise en charge permet de partitionner vos données dans un cluster Valkey ou Redis OSS (mode cluster activé).

Le schéma suivant illustre les différences entre un cluster Valkey ou Redis OSS (mode cluster désactivé) et un cluster Valkey ou Redis OSS (mode cluster activé).

Les clusters Valkey ou Redis OSS (mode cluster activé) prennent en charge la réplication via des partitions. L'opération API DescribeReplicationGroups(CLI : describe-replication-groups) répertorie les groupes de nœuds avec les nœuds membres, le rôle du nœud au sein du groupe de nœuds, ainsi que d'autres informations.

Lorsque vous créez un cluster Valkey ou Redis OSS, vous spécifiez si vous souhaitez créer un cluster avec le clustering activé. Les clusters Valkey ou Redis OSS (mode cluster désactivé) ne comportent jamais plus d'une partition, qui peut être redimensionnée horizontalement en ajoutant (jusqu'à cinq au total) ou en supprimant des nœuds de réplication en lecture. Pour plus d'informations, veuillez consulter Haute disponibilité avec les groupes de réplication, Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé) ou Suppression d'une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé). Les clusters Valkey ou Redis

OSS (mode cluster désactivé) peuvent également être redimensionnés verticalement en modifiant le type de nœud. Pour de plus amples informations, veuillez consulter Dimensionnement des nœuds de réplication pour Valkey ou Redis OSS (mode cluster désactivé).

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si le moteur est Valkey ou Redis OSS version 5.0.6 ou supérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Une fois qu'un cluster Valkey ou Redis OSS (mode cluster activé) est créé, il peut être modifié (redimensionné ou réduit). Pour plus d'informations, consultez Dimensionnement ElastiCache et Remplacement de nœuds (Valkey et Redis OSS).

Lorsque vous créez un nouveau cluster, vous pouvez l'alimenter avec des données de l'ancien cluster afin qu'il ne démarre pas vide. Cette approche fonctionne uniquement si le groupe de clusters a le même nombre de partitions que l'ancien cluster. Elle peut être utile si vous devez modifier le type de nœud ou la version de votre moteur. Pour plus d'informations, consultez Réalisation de sauvegardes manuelles et Restauration à partir d'une sauvegarde dans un nouveau cache.

Connexion aux nœuds

Connexion aux nœuds Valkey ou Redis OSS

Avant de tenter de vous connecter aux nœuds Valkey ou Redis OSS de votre cluster, vous devez disposer des points de terminaison des nœuds. Pour rechercher les points de terminaison, consultez les sections suivantes :

- Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)
- Recherche de points de terminaison pour un cluster Valkey ou Redis OSS (mode cluster activé) (console)
- Recherche des points de terminaison (AWS CLI)
- Recherche de points de terminaison (ElastiCache API)

Dans l'exemple suivant, vous utilisez l'utilitaire valkey-cli pour vous connecter à un cluster qui exécute Valkey ou Redis OSS.

> **Note**
>
> Pour plus d'informations sur les commandes disponibles, consultez la page Web Commandes.

Pour vous connecter à un cluster Valkey ou Redis OSS à l'aide du valkey-cli

1. Connectez-vous à votre EC2 instance Amazon à l'aide de l'utilitaire de connexion de votre choix.

   > **Note**
   >
   > Pour savoir comment vous connecter à une EC2 instance Amazon, consultez le guide de EC2 démarrage Amazon.

2. Pour construire `valkey-cli`, téléchargez et installez GNU Compiler Collection (`gcc`). À l'invite de commande de votre EC2 instance, entrez la commande suivante et saisissez-la `y` à l'invite de confirmation.

```
sudo yum install gcc
```

Une sortie similaire à la suivante s'affiche.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB     00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm                | 4.8 MB     00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm                  | 2.8 kB     00:00

...(output omitted)...

Complete!
```

3. Téléchargez et compilez l'utilitaire valkey-cli. Cet utilitaire est inclus dans la distribution du logiciel Valkey. À l'invite de commande de votre EC2 instance, tapez les commandes suivantes :

> **Note**
>
> Pour les systèmes Ubuntu, avant d'exécuter `make`, exécutez `make distclean`.

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make distclean      # ubuntu systems only
make
```

4. À l'invite de commande de votre EC2 instance, tapez la commande suivante.

```
src/valkey-cli -c -h mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

Une invite de commande Valkey ou Redis OSS similaire à la suivante s'affiche.

```
redis mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. Testez la connexion en exécutant les commandes Valkey ou Redis OSS.

   Vous êtes maintenant connecté au cluster et pouvez exécuter les commandes Valkey ou Redis OSS. Voici quelques exemples de commandes avec leurs réponses Valkey ou Redis OSS.

```
set a "hello"          // Set key "a" with a string value and no expiration
OK
get a                  // Get value for key "a"
"hello"
get b                  // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
get b
"Good-bye"
                    // wait 5 seconds
get b
(nil)                  // key has expired, nothing returned
quit                   // Exit from valkey-cli
```

Pour la connexion à des nœuds ou à des clusters avec un chiffrement SSL (en transit activé), consultez ElastiCache chiffrement en transit (TLS).

Connexion aux nœuds Memcached

Avant d'essayer de vous connecter à votre cluster Memcached, vous devez connaître les points de terminaison pour les nœuds. Pour rechercher les points de terminaison, consultez les sections suivantes :

- Trouver les points de terminaison d'un cluster (console) (Memcached)
- Recherche des points de terminaison (AWS CLI)
- Recherche de points de terminaison (ElastiCache API)

Dans l'exemple suivant, vous utilisez l'utilitaire telnet pour vous connecter à un nœud Memcached.

> **Note**
>
> Pour plus d'informations sur Memcached et les commandes Memcached disponibles, consultez le site web Memcached.

Pour se connecter à un nœud à l'aide telnet

1. Connectez-vous à votre EC2 instance Amazon à l'aide de l'utilitaire de connexion de votre choix.

   > **Note**
   >
   > Pour savoir comment vous connecter à une EC2 instance Amazon, consultez le guide de EC2 démarrage Amazon.

2. Téléchargez et installez l'utilitaire Telnet sur votre EC2 instance Amazon. À l'invite de commande de votre EC2 instance Amazon, tapez la commande suivante et tapez y à l'invite de commande.

   ```
   sudo yum install telnet
   ```

   Une sortie similaire à la suivante s'affiche.

   ```
   Loaded plugins: priorities, security, update-motd, upgrade-helper
   Setting up Install Process
   Resolving Dependencies
   --> Running transaction check

   ...(output omitted)...

   Total download size: 63 k
   Installed size: 109 k
   Is this ok [y/N]: y
   Downloading Packages:
   telnet-0.17-47.7.amzn1.x86_64.rpm                        |  63 kB     00:00

   ...(output omitted)...
   ```

```
Complete!
```

3. À l'invite de commande de votre EC2 instance Amazon, tapez la commande suivante, en remplaçant le point de terminaison de votre nœud par celui illustré dans cet exemple.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

Une sortie similaire à la suivante s'affiche.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Testez la connexion en exécutant des commandes Memcached.

Vous êtes actuellement connecté à un nœud et vous pouvez exécuter les commandes Memcached. Voici un exemple.

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello            // Set value as "hello"
STORED
get a            // Get value for key "a"
VALUE a 0 5
hello
END
get b            // Get value for key "b" results in miss
END
>
```

Types de nœuds pris en charge

ElastiCache prend en charge les types de nœuds suivants. En général, les types de la génération actuelle offrent davantage de mémoire et de puissance de calcul pour un coût inférieur, par rapport à leurs homologues équivalents de la génération précédente.

Pour plus d'informations sur les performances de chaque type de nœud, consultez Amazon EC2 Instance Types.

> **Note**
>
> Les types d'instances suivants sont pris en charge dans les régions AWS Asie-Pacifique (Thaïlande) et Mexique (centre) :
>
> - m7g/r7g : grand, xl, 2xl, 4xl, 8xl, 12xl et 16xl.
> - t3/t4g : micro, petit et moyen.

Pour plus d'informations sur la taille de nœud à utiliser, veuillez consulter Choix de la taille de votre nœud.

Rubriques

- Génération actuelle (Memcached)
- Génération actuelle (Valkey ou Redis OSS)
- Types de nœuds pris en charge par région AWS
- Instances à capacité extensible
- Informations connexes

Génération actuelle (Memcached)

Les tableaux suivants indiquent la bande passante de base et la bande passante en rafale pour les types d'instances qui utilisent le mécanisme I/O de crédit réseau pour dépasser leur bande passante de base.

> **Note**
>
> Les types d'instances dont les performances réseau sont stables utilisent un mécanisme I/O de crédit réseau pour dépasser au maximum leur bande passante de base.

Général

Type d'instance	Version Memcached minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1,876	12,5
cache.m7g.2xlarge		3,75	15
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22,5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1.5,16	0.75	10,0
cache.m6g.xlarge	1.5,16	1,25	10,0
cache.m6g.2xlarge	1.5,16	2,5	10,0
cache.m6g.4xlarge	1.5,16	5.0	10,0
cache.m6g.8xlarge	1.5,16	12	N/A
cache.m6g.12xlarge	1.5,16	20	N/A
cache.m6g.16xlarge	1.5,16	25	N/A
cache.m5.large	1.5,16	0.75	10,0

Type d'instance	Version Memcached minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.m5.xlarge	1.5,16	1,25	10,0
cache.m5.2xlarge	1.5,16	2,5	10,0
cache.m5.4xlarge	1.5,16	5.0	10,0
cache.m5.12xlarge	1.5,16	N/A	N/A
cache.m5.24xlarge	1.5,16	N/A	N/A
cache.m4.large	1.5,16	0,45	1.2
cache.m4.xlarge	1.5,16	0.75	2,8
cache.m4.2xlarge	1.5,16	1.0	10,0
cache.m4.4xlarge	1.5,16	2.0	10,0
cache.m4.10xlarge	1.5,16	5.0	10,0
cache.t4g.micro	1.5,16	0,064	5.0
cache.t4g. petit	1.5,16	0,128	5.0
cache.t4g.medium	1.5,16	0,256	5.0
cache.t3.micro	1.5,16	0,064	5.0
cache.t3.small	1.5,16	0,128	5.0
cache.t3.medium	1.5,16	0,256	5.0
cache.t2.micro	1.5,16	0,064	1,024
cache.t2.small	1.5,16	0,128	1,024
cache.t2.medium	1.5,16	0,256	1,024

Mémoire optimisée pour Memcached

Type d'instance	Version minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	N/A
cache.r7g.12xlarge		22,5	N/A
cache.r7g.16xlarge		30	N/A
cache.r6g.large	1.5,16	0.75	10,0
cache.r6g.xlarge	1.5,16	1,25	10,0
cache.r6g.2xlarge	1.5,16	2,5	10,0
cache.r6g.4xlarge	1.5,16	5.0	10,0
cache.r6g.8xlarge	1.5,16	12	N/A
cache.r6g.12xlarge	1.5,16	20	N/A
cache.r6g.16xlarge	1.5,16	25	N/A
cache.r5.large	1.5,16	0.75	10,0
cache.r5.xlarge	1.5,16	1,25	10,0
cache.r5.2xlarge	1.5,16	2,5	10,0
cache.r5.4xlarge	1.5,16	5.0	10,0
cache.r5.12xlarge	1.5,16	20	N/A

Type d'instance	Version minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r5.24xlarge	1.5,16	25	N/A
cache.r4.large	1.5,16	0.75	10,0
cache.r4.xlarge	1.5,16	1,25	10,0
cache.r4.2xlarge	1.5,16	2,5	10,0
cache.r4.4xlarge	1.5,16	5.0	10,0
cache.r4.8xlarge	1.5,16	12	N/A
cache.r4.16xlarge	1.5,16	25	N/A

Réseau optimisé pour Memcached

Type d'instance	Version minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.c7gn.large	1.6.6	6,25	30
cache.c7gn.xlarge	1.6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

Génération actuelle (Valkey ou Redis OSS)

Pour plus d'informations sur la génération précédente, reportez-vous à la section Nœuds de la génération précédente.

> **Note**
>
> Les types d'instances dont les performances réseau sont stables utilisent un mécanisme I/O de crédit réseau pour dépasser au maximum leur bande passante de base.

Général

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.m7g.large	6.2	N	N	N	0,937	12,5
cache.m7g.xlarge	6.2	Y	Y	Y	1,876	12,5
cache.m7g.2xlarge	6.2	Y	Y	Y	3,75	15
cache.m7g.4xlarge	6.2	Y	Y	Y	7,5	15
cache.m7g.8xlarge	6.2	Y	Y	Y	15	N/A
cache.m7g.12xlarge	6.2	Y	Y	Y	22,5	N/A

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.m7g.16xlarge	6.2	Y	Y	Y	30	N/A
cache.m6g.large	5.0.6	N	N	N	0.75	10,0
cache.m6g.xlarge	5.0.6	Y	Y	Y	1,25	10,0
cache.m6g.2xlarge	5.0.6	Y	Y	Y	2,5	10,0
cache.m6g.4xlarge	5.0.6	Y	Y	Y	5.0	10,0
cache.m6g.8xlarge	5.0.6	Y	Y	Y	12	N/A
cache.m6g.12xlarge	5.0.6	Y	Y	Y	20	N/A
cache.m6g.16xlarge	5.0.6	Y	Y	Y	25	N/A
cache.m5.large	3.2.4	N	N	N	0.75	10,0
cache.m5.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.m5.2xlarge	3.2.4	Y	Y	Y	2,5	10,0

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.m5.4xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.m5.12xlarge	3.2.4	Y	Y	Y	12	N/A
cache.m5.24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.m4.large	3.2.4	N	N	N	0,45	1.2
cache.m4.xlarge	3.2.4	Y	N	N	0.75	2,8
cache.m4.2xlarge	3.2.4	Y	Y	Y	1.0	10,0
cache.m4.4xlarge	3.2.4	Y	Y	Y	2.0	10,0
cache.m4.10xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.t4g.micro	3.2.4	N	N	N	0,064	5.0
cache.t4g. petit	5.0.6	N	N	N	0,128	5.0
cache.t4g.medium	5.0.6	N	N	N	0,256	5.0
cache.t3.micro	3.2.4	N	N	N	0,064	5.0

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.t3.small	3.2.4	N	N	N	0,128	5.0
cache.t3.medium	3.2.4	N	N	N	0,256	5.0
cache.t2.micro	3.2.4	N	N	N	0,064	1,024
cache.t2.small	3.2.4	N	N	N	0,128	1,024
cache.t2.medium	3.2.4	N	N	N	0,256	1,024

Mémoire optimisée

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r7g.large	6.2	N	N	N	0,937	12,5
cache.r7g.xlarge	6.2	Y	Y	Y	1,876	12,5
cache.r7g .2xlarge	6.2	Y	Y	Y	3,75	15

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r7g.4xlarge	6.2	Y	Y	Y	7,5	15
cache.r7g.8xlarge	6.2	Y	Y	Y	15	N/A
cache.r7g.12xlarge	6.2	Y	Y	Y	22,5	N/A
cache.r7g.16xlarge	6.2	Y	Y	Y	30	N/A
cache.r6g.large	5.0.6	N	N	N	0.75	10,0
cache.r6g.xlarge	5.0.6	Y	Y	Y	1,25	10,0
cache.r6g.2xlarge	5.0.6	Y	Y	Y	2,5	10,0
cache.r6g.4xlarge	5.0.6	Y	Y	Y	5.0	10,0
cache.r6g.8xlarge	5.0.6	Y	Y	Y	12	N/A
cache.r6g.12xlarge	5.0.6	Y	Y	Y	20	N/A
cache.r6g.16xlarge	5.0.6	Y	Y	Y	25	N/A

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r5.large	3.2.4	N	N	N	0.75	10,0
cache.r5.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.r5.2xlarge	3.2.4	Y	Y	Y	2,5	10,0
cache.r5.4xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.r5.12xlarge	3.2.4	Y	Y	Y	12	N/A
cache.r5.24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.r4.large	3.2.4	N	N	N	0.75	10,0
cache.r4.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.r4.2xlarge	3.2.4	Y	Y	Y	2,5	10,0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	N/A
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	N/A

Mémoire optimisée avec la hiérarchisation des données

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r6gd.xlarge	6.2.0	Y	N	N	1,25	10
cache.r6g.2xlarge	6.2.0	Y	Y	Y	2,5	10
cache.r6g.4xlarge	6.2.0	Y	Y	Y	5.0	10
cache.r6g.8xlarge	6.2.0	Y	Y	Y	12	N/A
cache.r6g.12xlarge	6.2.0	Y	Y	Y	20	N/A
cache.r6g.16xlarge	6.2.0	Y	Y	Y	25	N/A

Optimisés pour le réseau

Type d'instance	Version minimale de Redis OSS prise en charge	Amélioré I/O avec Redis OSS 5.0.6+	Déchargement TLS avec Redis OSS 6.2.5+	I/O Multiplexage amélioré avec Redis OSS 7.0.4+	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.c7gn.large	6.2	N	N	N	6,25	30
cache.c7gn.xlarge	6.2	Y	Y	Y	12,5	40
cache.c7gn.2xlarge	6.2	Y	Y	Y	25	50
cache.c7gn.4xlarge	6.2	Y	Y	Y	50	N/A
cache.c7gn.8xlarge	6.2	Y	Y	Y	100	N/A
cache.c7gn.12xlarge	6.2	Y	Y	Y	150	N/A
cache.c7gn.16xlarge	6.2	Y	Y	Y	200	N/A

Types de nœuds pris en charge par région AWS

Les types de nœuds pris en charge peuvent varier d'une AWS région à l'autre. Pour plus de détails, consultez les ElastiCachetarifs Amazon.

Instances à capacité extensible

Vous pouvez lancer des nœuds de cache T4g, T3-Standard et T2-Standard à usage général sur Amazon. ElastiCache Ces nœuds fournissent un niveau de base de performance de processeur

avec la possibilité d'étendre l'utilisation du processeur à tout moment jusqu'à épuisement des crédits accumulés. Un crédit UC fournit la capacité d'un cœur UC complet pendant une minute.

Les nœuds ElastiCache T4g, T3 et T2 d'Amazon sont configurés en standard et adaptés aux charges de travail dont l'utilisation moyenne du processeur est constamment inférieure aux performances de base de l'instance. Pour émettre en rafales au-dessus du niveau de base, l'instance dépense les crédits cumulés dans son solde de crédits UC. Si le nœud s'exécute faiblement sur les crédits accumulés, les performances sont progressivement réduites vers le niveau de performance de référence. Cet affaiblissement progressif garantit que le nœud ne se heure pas à une baisse conséquente des performance lorsque le solde de crédits UC cumulés est bas. Pour plus d'informations, consultez la section Crédits CPU et performances de base pour les instances Burstable Performance dans le guide de l' EC2 utilisateur Amazon.

Le tableau suivant recense les types de nœud aux performances à capacité extensible et la fréquence à laquelle les crédits UC sont obtenus par heure. Il indique également le nombre maximum de crédits CPU gagnés qu'un nœud peut accumuler et le nombre de v CPUs par nœud. En outre, il fournit le niveau de performance de référence comme pourcentage d'une performance de cœur complet (à l'aide d'un seul processeur virtuel).

Crédits UC gagnés par heure	Maximum de crédits gagnés pouvant être accumulés *	v CPUs	Performances de base par vCPU	Mémoire (Gio)	Performances réseau
12	288	2	10 %	0.5	Jusqu'à 5 gigabits
24	576	2	20 %	1,37	Jusqu'à 5 gigabits
24	576	2	20 %	3,09	Jusqu'à 5 gigabits
12	288	2	10 %	0.5	Jusqu'à 5 gigabits

Crédits UC gagnés par heure	Maximum de crédits gagnés pouvant être accumulés *	v CPUs	Performances de base par vCPU	Mémoire (Gio)	Performances réseau
24	576	2	20 %	1,37	Jusqu'à 5 gigabits
24	576	2	20 %	3,09	Jusqu'à 5 gigabits
6	144	1	10 %	0.5	Faible à modérée
12	288	1	20 %	1,55	Faible à modérée
24	576	2	20 %	3,22	Faible à modérée

* Le nombre de crédits pouvant être accumulés est équivalent au nombre de crédits pouvant être gagnés en 24 heures.

** Les performances de base du tableau sont indiquées par vCPU. Certaines tailles de nœud qui ont plus d'un processeur virtuel. Pour cela, calculez l'utilisation du processeur de référence pour le nœud en multipliant le pourcentage de vCPU par le nombre de v. CPUs

Les métriques de crédit UC suivantes sont disponibles pour vos instances T3 et T4g de performance à capacité extensible :

> **Note**
>
> Ces métriques ne sont pas disponibles pour les instances de performance à capacité extensible T2.

- `CPUCreditUsage`
- `CPUCreditBalance`

Pour de plus amples informations sur ces métriques, consultez Métriques de crédits d'UC.

En outre, sachez que :

- Tous les types de nœud de génération courante sont créés dans un Virtual Private Cloud (VPC) basé sur Amazon VPC par défaut.
- Les fichiers d'ajout uniquement (AOF) Redis OSS ne sont pas pris en charge pour les instances T2. Les variables de configuration Redis OSS `appendfsync` ne sont pas prises en charge sur Redis OSS version 2.8.22 `appendonly` et versions ultérieures.

Informations connexes

- Caractéristiques et détails des ElastiCache produits Amazon
- Paramètres spécifiques au type de nœud Memcached pour Memcached
- Paramètres OSS Valkey et Redis
- Chiffrement en transit (TLS)

Redémarrage des nœuds

Certaines modifications nécessitent le redémarrage d'un cluster Valkey, Memcached ou Redis OSS pour que les modifications soient appliquées. Par exemple, pour certains paramètres, la modification de la valeur du paramètre dans un groupe de paramètres est appliquée uniquement après un redémarrage.

Rubriques

- Redémarrage des nœuds Redis OSS (mode cluster désactivé uniquement)
- Redémarrer un cluster pour Memcached

Redémarrage des nœuds Redis OSS (mode cluster désactivé uniquement)

Pour les clusters Valkey ou Redis OSS (mode cluster désactivé), les paramètres des groupes de paramètres appliqués uniquement après le redémarrage sont les suivants :

- activerehachage
- bases de données

Les nœuds Redis ne peuvent être mis à jour que via la ElastiCache console. Vous ne pouvez redémarrer qu'un seul nœud à la fois. Pour redémarrer plusieurs nœuds, vous devez répéter le processus pour chaque nœud.

> **Modifications des paramètres Valkey ou Redis OSS (mode cluster activé)**
>
> Si vous modifiez les paramètres suivants sur un cluster Valkey ou Redis OSS (mode cluster activé), suivez les étapes suivantes.
>
> - activerehachage
> - bases de données
>
> 1. Créez une sauvegarde manuelle de votre cluster. Consultez Réalisation de sauvegardes manuelles.
> 2. Supprimez le cluster Valkey ou Redis OSS (mode cluster activé). Consultez Supprimer un cluster dans ElastiCache.
> 3. Restaurez le cluster grâce au groupe de paramètres modifiés et à la sauvegarde pour alimenter le nouveau cluster. Consultez Restauration à partir d'une sauvegarde dans un nouveau cache.
>
> Les modifications apportées à d'autres paramètres ne nécessitent pas cela.

En utilisant le AWS Management Console

Vous pouvez redémarrer un nœud à l'aide de la ElastiCache console.

Pour redémarrer un nœud (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région qui s'applique.
3. Dans le volet de navigation de gauche, choisissez Redis OSS.

La liste des clusters exécutant Redis OSS apparaît.

4. Choisissez le cluster sous Cluster Name (Nom du cluster).
5. Sous Node name (Nom du nœud), sélectionnez la case d'option en regard du nœud que vous souhaitez redémarrer.
6. Choisissez Actions, puis Reboot node (Redémarrer le nœud).

Pour redémarrer plusieurs clusters, répétez les étapes 2 à 5 pour chaque nœud que vous voulez redémarrer. Vous n'avez pas besoin d'attendre que le redémarrage d'un nœud soit terminé pour en redémarrer un autre.

Redémarrer un cluster pour Memcached

Lorsque vous redémarrez un cluster Memcached, le cluster vide toutes ses données et redémarre son moteur. Au cours de ce processus, vous ne pouvez pas accéder au cluster. Etant donné que le cluster a vidé toutes ses données, lorsque le cluster est disponible à nouveau, il est vide.

Vous pouvez redémarrer un cluster à l'aide de la ElastiCache console, de AWS CLI, ou de l' ElastiCache API. Que vous utilisiez la ElastiCache console, l'API AWS CLI ou l' ElastiCache API, vous ne pouvez lancer le redémarrage que d'un seul cluster. Pour redémarrer plusieurs clusters, vous devez répéter l'opération sur le processus ou l'action.

En utilisant le AWS Management Console

Vous pouvez redémarrer un cluster à l'aide de la ElastiCache console.

Pour redémarrer un cluster (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste en haut à droite, choisissez la AWS région qui vous intéresse.
3. Dans le volet de navigation, choisissez le moteur s'exécutant sur le cluster que vous voulez redémarrer.

   La liste des clusters exécutant le moteur choisi s'affiche.
4. Choisissez le cluster à redémarrer dans la zone située à gauche du nom du cluster.

   Le bouton Redémarrer devient actif.

Si vous choisissez plusieurs clusters, le bouton Redémarrer n'est pas actif.

5. Choisissez Redémarrer.

   L'écran de confirmation de redémarrage du cluster de s'affiche.

6. Pour redémarrer le cluster, choisissez Reboot. L'état du cluster devient redémarrage des nœuds de cluster.

   Pour ne pas redémarrer le cluster, choisissez Cancel.

Pour redémarrer plusieurs clusters, répétez les étapes 2 à 5 pour chaque cluster que vous voulez redémarrer. Vous n'avez pas besoin d'attendre que le redémarrage d'un cluster soit terminé pour en redémarrer un autre.

Pour redémarrer un nœud spécifique, sélectionnez le nœud, puis choisissez Reboot (Redémarrer).

En utilisant le AWS CLI

Pour redémarrer un cluster (AWS CLI), utilisez l'opération `reboot-cache-cluster` de la CLI.

Pour redémarrer des nœuds de clusters spécifiques, utilisez le paramètre `--cache-node-ids-to-reboot` pour afficher les clusters spécifiques à redémarrer. La commande suivante redémarre les nœuds 0001, 0002 et 0004 de my-cluster.

Pour Linux, macOS ou Unix :

```
aws elasticache reboot-cache-cluster \
    --cache-cluster-id my-cluster \
    --cache-node-ids-to-reboot 0001 0002 0004
```

Pour Windows :

```
aws elasticache reboot-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --cache-node-ids-to-reboot 0001 0002 0004
```

Pour redémarrer tous les nœuds du cluster, utilisez le paramètre `--cache-node-ids-to-reboot` et répertoriez tous les ID de nœud du cluster. Pour de plus amples informations, veuillez consulter reboot-cache-cluster.

Utilisation de l' ElastiCache API

Pour redémarrer un cluster à l'aide de l' ElastiCache API, utilisez l'`RebootCacheCluster`action.

Pour redémarrer des nœuds de clusters spécifiques, utilisez le paramètre `CacheNodeIdsToReboot` pour afficher les clusters spécifiques à redémarrer. La commande suivante redémarre les nœuds 0001, 0002 et 0004 de my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=RebootCacheCluster
   &CacheClusterId=my-cluster
   &CacheNodeIdsToReboot.member.1=0001
   &CacheNodeIdsToReboot.member.2=0002
   &CacheNodeIdsToReboot.member.3=0004
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Pour redémarrer tous les nœuds du cluster, utilisez le paramètre `CacheNodeIdsToReboot` et répertoriez tous les ID de nœud du cluster. Pour de plus amples informations, veuillez consulter RebootCacheCluster.

Remplacement de nœuds (Valkey et Redis OSS)

Amazon met ElastiCache fréquemment à niveau son parc en appliquant des correctifs et des mises à niveau aux instances de manière fluide. Cependant, nous devons parfois relancer vos ElastiCache nœuds 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 cache de remplacement générés automatiquement par Amazon ElastiCache peuvent avoir des adresses IP différentes. Vous êtes chargé d'examiner la configuration de votre application pour vous assurer que vos nœuds de cache sont associés aux adresses IP appropriées.

La liste suivante identifie les actions que vous pouvez effectuer lorsque vous ElastiCache planifiez le remplacement de l'un de vos nœuds Valkey ou Redis OSS. Afin d'accélérer la recherche des informations dont vous avez besoin, choisissez dans le menu suivant.

- Do nothing— Laissez Amazon ElastiCache remplacer le nœud comme prévu.
- Change your maintenance window – changer pour un créneau de maintenance plus approprié.
- Configurations Valkey ou Redis OSS (mode cluster activé)
  - Replace the only node in any Valkey or Redis OSS cluster— Procédure permettant de remplacer un nœud dans un cluster Valkey ou Redis OSS à l'aide de la sauvegarde et de la restauration.
  - Replace a replica node in any Valkey or Redis OSS cluster— Procédure permettant de remplacer une réplique en lecture dans n'importe quel cluster Valkey ou Redis OSS en augmentant ou en diminuant le nombre de répliques sans interruption du cluster.
  - Replace any node in a Valkey or Redis OSS (cluster mode enabled) shard— Procédure dynamique sans interruption de service du cluster pour remplacer un nœud dans un cluster

Valkey ou Redis OSS (mode cluster activé) par une mise à l'échelle externe et une mise à l'échelle interne.

- Configurations Valkey ou Redis OSS (mode cluster désactivé)
  - Replace the only node in any Valkey or Redis OSS cluster— Procédure pour remplacer n'importe quel nœud d'un cluster Valkey ou Redis OSS à l'aide de la sauvegarde et de la restauration.
  - Replace a replica node in any Valkey or Redis OSS cluster— Procédure permettant de remplacer une réplique en lecture dans n'importe quel cluster Valkey ou Redis OSS en augmentant ou en diminuant le nombre de répliques sans interruption du cluster.
  - Replace a node in a Valkey or Redis OSS (cluster mode disabled) cluster— Procédure de remplacement d'un nœud dans un cluster Valkey ou Redis OSS (mode cluster désactivé) par réplication.
  - Replace a Valkey or Redis OSS (cluster mode disabled) read-replica— Procédure permettant de remplacer manuellement une réplique en lecture dans un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé).
  - Replace a Valkey or Redis OSS (cluster mode disabled) primary node— Procédure permettant de remplacer manuellement le nœud principal dans un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé).

Options de remplacement des nœuds Valkey et Redis OSS

- Ne rien faire : si vous ne faites rien, ElastiCache remplace le nœud comme prévu.

Pour les configurations hors cluster avec le basculement automatique activé, les clusters sur Valkey 7.2 et versions ultérieures et Redis OSS 5.0.6 et versions ultérieures sont complètement remplacés tandis que le cluster continue de rester en ligne et de traiter les demandes d'écriture entrantes. Pour les clusters compatibles avec le basculement automatique sur Redis OSS 4.0.10 ou version antérieure, vous remarquerez peut-être une brève interruption d'écriture de quelques secondes maximum associée aux mises à jour DNS.

Si le nœud est membre d'un cluster activé ElastiCache pour le basculement automatique, Valkey ou Redis OSS améliore la disponibilité lors des correctifs, des mises à jour et des autres remplacements de nœuds liés à la maintenance.

Pour les configurations de ElastiCache cluster configurées pour être utilisées ElastiCache pour les clients de cluster Valkey ou Redis OSS, le remplacement est désormais terminé pendant que le cluster traite les demandes d'écriture entrantes.

Pour les configurations hors cluster avec le basculement automatique activé, les clusters sur Valkey 7.2 et versions ultérieures et Redis OSS 5.0.6 et versions ultérieures sont complètement remplacés tandis que le cluster continue de rester en ligne et de traiter les demandes d'écriture entrantes. Pour les clusters compatibles avec le basculement automatique sur Redis OSS 4.0.10 ou version antérieure, vous remarquerez peut-être une brève interruption d'écriture de quelques secondes maximum associée aux mises à jour DNS.

Si le nœud est autonome, Amazon lance d' ElastiCache abord un nœud de remplacement, puis se synchronise à partir du nœud existant. Le nœud existant ne sera pas disponible pour les demandes de service pendant cette période. Une fois la synchronisation terminée, le nœud existant est arrêté et le nouveau nœud prend sa place. ElastiCache fait de son mieux pour conserver vos données au cours de cette opération.

- Modifiez votre fenêtre de maintenance — Pour les événements de maintenance planifiés, vous recevez un e-mail ou une notification de la part de ElastiCache. 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 les ressources suivantes :
  - Modification d'un ElastiCache cluster
  - Modification d'un groupe de réplication

> **Note**
>
> La possibilité de modifier votre fenêtre de remplacement en déplaçant votre fenêtre de maintenance n'est disponible que lorsque la ElastiCache notification 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 reportez votre fenêtre de maintenance au mercredi à 16 h 00 (plus tôt dans le semaine que la date et l'heure actuelles). Le nœud est remplacé le mercredi 15 novembre à 16 h 00.

Pour obtenir des instructions, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

- Remplacez le seul nœud d'un cluster Valkey ou Redis OSS : si le cluster ne possède aucune réplique de lecture, vous pouvez utiliser la procédure suivante pour remplacer le nœud.

  Pour remplacer le seul nœud en utilisant la sauvegarde et la restauration

  1. Créez un instantané du cluster du nœud. Pour obtenir des instructions, veuillez consulter Réalisation de sauvegardes manuelles.
  2. Créez un nouveau cluster à partir de l'instantané. Pour obtenir des instructions, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.
  3. Supprimez le cluster avec le nœud qui doit être remplacé. Pour obtenir des instructions, veuillez consulter Supprimer un cluster dans ElastiCache.
  4. Dans votre application, remplacez le point de terminaison de l'ancien nœud par celui du nouveau nœud.

- Remplacez un nœud de réplique dans n'importe quel cluster Valkey ou Redis OSS : pour remplacer un cluster de répliques, augmentez le nombre de répliques. Pour ce faire, ajoutez un réplica, puis diminuez le nombre de réplicas en supprimant le réplica à remplacer. Il s'agit d'un processus dynamique sans temps d'arrêt du cluster.

> **Note**
>
> Si votre partition ou groupe de réplication comporte déjà cinq réplicas, vous devez inverser les étapes 1 et 2.

Pour remplacer une réplique dans n'importe quel cluster Valkey ou Redis OSS

1. Augmentez le nombre de réplicas en ajoutant un réplica à la partition ou au groupe de réplication. Pour de plus amples informations, veuillez consulter Augmentation du nombre de réplicas dans une partition.
2. Supprimez le réplica que vous souhaitez remplacer. Pour de plus amples informations, veuillez consulter Diminution du nombre de réplicas dans une partition.
3. Mettez à jour les points de terminaison dans votre application.

- Remplacez n'importe quel nœud d'une partition Valkey ou Redis OSS (mode cluster activé) : pour remplacer le nœud d'un cluster sans interruption de service, utilisez le repartage en ligne. Ajoutez d'abord une partition à l'aide d'une mise à l'échelle supérieure, puis supprimez la partition avec le nœud à remplacer avec une mise à l'échelle inférieure.

  Pour remplacer n'importe quel nœud d'un cluster Valkey ou Redis OSS (mode cluster activé)

  1. Monter en charge : Ajoutez une partition supplémentaire avec la même configuration que la partition existante avec le nœud à remplacer. Pour de plus amples informations, veuillez consulter Ajout de partitions avec le repartitionnement en ligne.
  2. Mettre à l'échelle : Supprimez la partition avec le nœud à remplacer. Pour de plus amples informations, veuillez consulter Suppression de partitions avec le repartitionnement en ligne.
  3. Mettez à jour les points de terminaison dans votre application.

- Remplacer un nœud dans un cluster Valkey ou Redis OSS (mode cluster désactivé) — Si le cluster est un cluster Valkey ou Redis OSS (mode cluster désactivé) sans aucune réplique de lecture, suivez la procédure suivante pour remplacer le nœud.

Remplacer le nœud en utilisant la réplication (mode cluster désactivé uniquement)

1. Ajoutez une réplication au cluster en définissant le nœud de remplacement prévu comme nœud primaire. N'activez pas le mode Multi-AZ sur ce cluster. Pour obtenir des instructions, veuillez consulter Pour ajouter une réplication à un cluster Valkey ou Redis OSS sans partitions.
2. Ajoutez un réplica en lecture au cluster. Pour obtenir des instructions, veuillez consulter Pour ajouter des nœuds à un ElastiCache cluster (console).
3. Promouvez le dernier réplica en lecture créé au rang de réplica principal. Pour obtenir des instructions, veuillez consulter Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).
4. Supprimer le nœud qui doit être remplacé. Pour obtenir des instructions, veuillez consulter Supprimer des nœuds d'un ElastiCache cluster.
5. Dans votre application, remplacez le point de terminaison de l'ancien nœud par celui du nouveau nœud.

- Remplacer une réplique en lecture Valkey ou Redis OSS (mode cluster désactivé) : si le nœud est une réplique en lecture, remplacez-le.

  Si votre cluster possède un seul nœud de réplica et que le mode Multi-AZ est activé, vous devez désactiver ce dernier avant de supprimer le réplica. Pour obtenir des instructions, veuillez consulter Modification d'un groupe de réplication.

  Pour remplacer un Valkey ou un Redis OSS (mode cluster désactivé), lisez la réplique

  1. Supprimez le réplica qui doit être remplacé. Pour obtenir des instructions, veuillez consulter les sections suivantes :
     - Diminution du nombre de réplicas dans une partition
     - Supprimer des nœuds d'un ElastiCache cluster
  2. Ajoutez un nouveau réplica pour remplacer celui qui doit être remplacé. Si vous utilisez le même nom que le réplica que vous venez de supprimer, vous pouvez ignorer l'étape 3. Pour obtenir des instructions, veuillez consulter les sections suivantes :
     - Augmentation du nombre de réplicas dans une partition
     - Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé)

3. Dans votre application, remplacez le point de terminaison de l'ancien réplica par celui du nouveau réplica.
4. Si vous avez désactivé le mode Multi-AZ au début, réactivez-le. Pour obtenir des instructions, veuillez consulter Activation du multi-AZ .

- Remplacez un nœud principal Valkey ou Redis OSS (mode cluster désactivé) : si le nœud est le nœud principal, commencez par promouvoir une réplique en lecture en nœud principal. Supprimez ensuite le réplica qui était le nœud primaire.

  Si votre cluster possède un seul réplica et que le mode Multi-AZ est activé, vous devez désactiver ce dernier avant de supprimer le réplica dans l'étape 2. Pour obtenir des instructions, veuillez consulter Modification d'un groupe de réplication.

  Pour remplacer un nœud principal Valkey ou Redis OSS (mode cluster désactivé)

  1. Promouvez un réplica en lecture en réplica principal. Pour obtenir des instructions, veuillez consulter Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).
  2. Supprimez le nœud qui doit être remplacé (ancienne instance principale). Pour obtenir des instructions, veuillez consulter Supprimer des nœuds d'un ElastiCache cluster.
  3. Ajoutez un nouveau réplica pour remplacer celui qui doit être remplacé. Si vous utilisez le même nom que le nœud que vous venez de supprimer, vous pouvez ignorer la modification des points de terminaison dans votre application.

     Pour obtenir des instructions, veuillez consulter Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé).
  4. Dans votre application, remplacez le point de terminaison de l'ancien nœud par celui du nouveau nœud.
  5. Si vous avez désactivé le mode Multi-AZ au début, réactivez-le. Pour obtenir des instructions, veuillez consulter Activation du multi-AZ .

Remplacement de nœuds (Memcached)

Amazon ElastiCache for Memcached met fréquemment à niveau son parc en appliquant des correctifs et des mises à niveau aux instances de manière fluide. Cependant, nous devons parfois relancer vos nœuds Memcached ElastiCache 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 cache de remplacement générés automatiquement par Amazon ElastiCache peuvent avoir des adresses IP différentes. Vous êtes chargé d'examiner la configuration de votre application pour vous assurer que vos nœuds de cache sont associés aux adresses IP appropriées.

La liste suivante identifie les actions que vous pouvez effectuer lorsque vous ElastiCache planifiez le remplacement de l'un de vos nœuds Memcached.

- Ne rien faire : si vous ne faites rien, ElastiCache remplace le nœud comme prévu. Lorsque le nœud est ElastiCache automatiquement remplacé par un nouveau nœud, le nouveau nœud est initialement vide.
- Modifiez votre fenêtre de maintenance — Pour les événements de maintenance planifiés, vous recevez un e-mail ou une notification de la part deElastiCache. 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 de plus amples informations, veuillez consulter Modification d'un ElastiCache cluster.

> **Note**
>
> La possibilité de modifier votre fenêtre de remplacement en déplaçant votre fenêtre de maintenance n'est disponible que lorsque la ElastiCache notification 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 reportez votre fenêtre de maintenance au mercredi à 16 h 00 (plus tôt dans le 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 des ElastiCache clusters.

- Manually replace the node (Remplacer manuellement le nœud) — si vous avez besoin de remplacer le nœud avant le prochain créneau de maintenance, remplacez-le manuellement.

  Si vous remplacez manuellement le nœud, les clés sont redistribuées. Cette redistribution entraîne des échecs d'accès au cache.

  Pour remplacer manuellement un nœud Memcached

  1. Supprimer le nœud qui doit être remplacé. Pour obtenir des instructions, veuillez consulter Supprimer des nœuds d'un ElastiCache cluster.
  2. Ajouter un nouveau nœud au cluster. Pour obtenir des instructions, veuillez consulter Ajouter des nœuds à un ElastiCache cluster.
  3. Si vous n'utilisez pas la découverte automatique sur ce cluster, accédez à votre application et remplacez chaque instance de point de terminaison de l'ancien nœud par le point de terminaison du nouveau nœud.

Nœuds réservés

La réservation d'un ou de plusieurs ElastiCache nœuds peut vous permettre de réduire les coûts. Des frais initiaux sont facturés pour les nœuds réservés en fonction du type de nœud et de la durée de réservation : un ou trois ans.

Pour voir si des nœuds réservés sont rentables pour vos cas d'utilisation, déterminez d'abord la taille du nœud et le nombre de nœuds dont vous avez besoin. Estimez ensuite l'utilisation du nœud et comparez le coût total d'utilisation de nœuds à la demande par rapport aux nœuds réservés. Vous pouvez combiner et associer des nœuds réservés et à la demande dans vos clusters. Pour plus d'informations sur les tarifs, consultez Amazon ElastiCache Pricing.

Rubriques

- Gestion des coûts avec des nœuds réservés
- Offres relatives aux nœuds réservés standards
- Dimensionnez les nœuds réservés flexibles
- Supprimer un nœud réservé
- Offres existantes relatives aux nœuds réservés
- Obtenir des informations sur des offres de nœuds réservés
- Achat d'un nœud réservé
- Obtenir des informations sur vos nœuds réservés

Gestion des coûts avec des nœuds réservés

La réservation d'un ou plusieurs nœuds peut vous permettre de réduire les coûts. Des frais initiaux sont facturés pour les nœuds réservés en fonction du type de nœud et de la durée de réservation : un ou trois ans. Ces frais sont largement inférieurs aux frais d'utilisation horaire facturés pour les nœuds à la demande.

Pour voir si des nœuds réservés sont rentables pour vos cas d'utilisation, déterminez d'abord la taille du nœud et le nombre de nœuds dont vous avez besoin. Estimez ensuite l'utilisation du nœud et comparez le coût total d'utilisation de nœuds à la demande par rapport aux nœuds réservés. Vous pouvez combiner et associer des nœuds réservés et à la demande dans vos clusters. Pour plus d'informations sur les tarifs, consultez Amazon ElastiCache Pricing.

AWS La région, le type de nœud et la durée du terme doivent être choisis lors de l'achat et ne peuvent pas être modifiés ultérieurement.

Vous pouvez utiliser l'API AWS Management Console AWS CLI, le ou l' ElastiCache API pour répertorier et acheter les offres de nœuds réservés disponibles.

Pour plus d'informations sur les nœuds réservés, consultez Amazon ElastiCache Reserved Nodes.

Offres relatives aux nœuds réservés standards

Lorsque vous achetez une instance de nœud réservé (RI) sur Amazon ElastiCache, vous pouvez vous engager à bénéficier d'un tarif réduit sur un type d'instance de nœud et une AWS région spécifiques pendant toute la durée de l'instance de nœud réservé. Pour utiliser une instance de nœud ElastiCache réservé Amazon, vous devez créer une nouvelle instance de ElastiCache nœud, comme vous le feriez pour une instance à la demande.

Si les spécifications de la nouvelle instance de nœud de réserve correspondent à une instance de nœud réservé existante pour votre compte, vous êtes facturé au tarif réduit proposé pour l'instance de nœud réservé. Dans le cas contraire, l'instance de nœud est facturée selon le tarif à la demande. Ces normes RIs sont disponibles à partir des familles d'instances R5 et M5.

> **Note**
>
> Tous les types d'offres décrits ci-dessous sont disponibles pour des durées d'un an et de trois ans.

Types d'offres

Aucun RI initial ne permet d'accéder à une ElastiCache instance réservée sans exiger de paiement initial. Votre ElastiCache instance réservée No Upfront facture un tarif horaire réduit pour chaque heure pendant le terme, quelle que soit l'utilisation.

Le RI initial partiel nécessite le paiement initial d'une partie de l' ElasticCache instance réservée. Les heures restantes pendant la période sont facturées à un taux réduit, quelle que soit l'utilisation. Cette option remplace l'option Utilisation intensive existante, qui est expliquée dans la section suivante.

Tous les frais initiaux RI exigent que le paiement intégral soit effectué au début du mandat du RI. Vous ne supportez aucun autre coût pour le reste du trimestre, quel que soit le nombre d'heures utilisées.

Dimensionnez les nœuds réservés flexibles

Tous les nœuds réservés sont flexibles en termes de taille. Lorsque vous achetez un nœud réservé, vous devez notamment spécifier le type de nœud, par exemple cache.r6g.xlarge. Pour plus d'informations sur les types de nœuds, consultez Amazon ElastiCache Pricing.

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, quelle que soit leur 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 cache.r6g.xlarge peut s'appliquer à un cache.r6g.2xlarge, mais pas à un cache.r6gd.large, car cache.r6g et cache.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

Mise à niveau des nœuds de Redis OSS vers Valkey

Avec le lancement de Valkey en ElastiCache, vous pouvez désormais appliquer votre réduction sur le nœud réservé Redis OSS au moteur de cache Valkey. Vous pouvez passer de Redis OSS à Valkey tout en continuant à bénéficier des contrats et des réservations existants. En plus de pouvoir appliquer vos avantages au sein de la famille et du moteur de nœuds de cache, vous pouvez même bénéficier d'une valeur incrémentielle accrue. Le prix de Valkey est réduit de 20 % par rapport à celui de Redis OSS, et grâce à la flexibilité des nœuds réservés, vous pouvez utiliser vos nœuds réservés Redis OSS pour couvrir 20 % de nœuds Valkey en cours d'exécution supplémentaires.

Pour calculer le taux réduit, chaque combinaison de ElastiCache nœud et de moteur possède un facteur de normalisation mesuré en unités. Les unités de nœuds réservés peuvent être appliquées à n'importe quel nœud en cours d'exécution au sein de la famille d'instances du nœud réservé pour un moteur donné. Les nœuds réservés Redis OSS peuvent également s'appliquer à tous les moteurs pour couvrir les nœuds Valkey en cours d'exécution. Le prix de Valkey étant réduit par rapport à Redis OSS et Memcached, ses unités pour un type d'instance donné sont inférieures, ce qui permet à un nœud réservé Redis OSS de couvrir un plus grand nombre de nœuds Valkey.

Par exemple, supposons que vous ayez acheté un nœud réservé pour un cache.r7g.4xlarge pour le moteur Redis OSS (32 unités) et que vous exécutiez un nœud Redis OSS cache.r7g.4xlarge (32

unités). Si vous mettez à niveau le nœud vers Valkey, le facteur de normalisation du nœud actif tombe à 25,6 unités, et votre nœud réservé existant vous fournit 6,4 unités supplémentaires à utiliser contre tout autre nœud Valkey ou Redis OSS en cours d'exécution de la famille cache.r7g dans la région. Vous pouvez l'utiliser pour couvrir 25 % d'un autre nœud Valkey cache.r7g.4xlarge du compte (25,6 unités) ou 100 % d'un nœud Valkey cache.r7g.xlarge (6,4 unités).

Comparaison de l'utilisation avec les unités normalisées

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

Taille de nœud	Unités normalisées avec Redis OSS ou Memcached	Unités normalisées avec Valkey
micro	0.5	0.4
petit	1	8.
medium	2	1.6
large	4	3.2
xlarge	8	6.4
2xlarge	16	12,8
4xlarge	32	25,6
6xlarge	48	38,4
8xlarge	64	51,2
10xlarge	80	64
12xlarge	96	76,8
16xlarge	128	102,4

Taille de nœud	Unités normalisées avec Redis OSS ou Memcached	Unités normalisées avec Valkey
24xlarge	192	153,6

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

Sinon, si vous avez une instance cache.r6gd.2xlarge 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).

Offres existantes relatives aux nœuds réservés

Il existe trois niveaux de réserves de nœuds héritées : utilisation intensive, utilisation moyenne et utilisation légère. Les nœuds peuvent être réservés quel que soit le niveau d'utilisation pendant un ou trois ans. Le type de nœud, le niveau d'utilisation et la durée de réservation influent sur vos coûts totaux. Avant d'acheter des nœuds réservés, vérifiez les économies que votre entreprise pourrait réaliser grâce à eux en comparant plusieurs modèles.

Les nœuds achetés pour une durée ou un niveau d'utilisation bien précis ne peuvent pas être modifiés.

Niveaux d'utilisation

Les nœuds de réservés pour utilisation intensive permettent les charges de travail qui ont une référence cohérente des capacités ou exécutent les charges de travail régulières. Ils nécessitent un engagement préalable plus élevé, mais si vous prévoyez d'exécuter plus de 79 % de la durée du nœud réservé, vous pouvez réaliser des économies plus importantes (jusqu'à 70 % à partir du prix à la demande). Avec les nœuds réservés à forte utilisation, vous ne payez qu'une seule fois. Il est ensuite suivi d'un tarif horaire inférieur pour la durée du contrat, que votre nœud soit en cours d'exécution ou non.

Les nœuds de réservés pour utilisation moyenne sont la meilleure option si vous prévoyez d'utiliser vos nœuds réservés une grande partie du temps et si vous souhaitez soit une redevance unique moins élevée, soit cesser de payer pour votre nœud lorsque vous arrêtez celui-ci. Ils constituent une option plus rentable si vous prévoyez d'exécuter plus de 40 % de la durée des nœuds réservés. Cette option peut vous faire économiser jusqu'à 64 % du prix à la demande. Avec les nœuds réservés pour une utilisation moyenne, vos frais ponctuels sont légèrement plus élevés qu'avec des nœuds réservés pour une utilisation faible, et vous bénéficiez de frais d'utilisation horaire inférieurs lorsque vous exécutez un nœud.

Les nœuds de réservés pour utilisation faible conviennent aux charges de travail Spot qui ont lieu uniquement une ou deux heures par jour ou quelques jours par semaine. Dans le cadre des nœuds de réservés pour utilisation faible, vous payez une cotisation unique accompagnée d'un taux d'utilisation horaire réduit lorsque votre nœud de est en cours d'exécution. Vous commencez à faire des économies lorsque votre nœud fonctionne à plus de 17 % du terme réservé du nœud. Vous

pouvez économiser jusqu'à 56 % sur les tarifs à la demande pendant tout le terme de votre nœud réservé.

Offres existantes relatives aux nœuds réservés

Offre	Coût initial	Coût d'utilisation	Avantages
Utilisation intensive	Le plus élevé	Plus faible coût horaire. S'applique à toute la période que vous utilisiez ou non le nœud de réservé.	Coût global plus faible si vous envisagez d'exécuter vos nœuds réservés plus de 79 % sur une durée de trois ans.
Utilisation moyenne	Moyen	Frais d'utilisation horaire facturés pour chaque heure d'exécution du nœud. Aucun frais horaire n'est facturé lorsque le nœud ne s'exécute pas.	Adapté aux charges de travail élastiques ou lorsque que vous envisagez une utilisation modérée, et plus de 40 % d'une durée de trois ans.
Utilisation légère	Le plus faible	Frais d'utilisation horaire facturés pour chaque heure d'exécution du nœud. Aucun frais horaire n'est facturé lorsque le nœud ne s'exécute pas. Le tarif horaire le plus haut de tous les types d'offres, mais il s'applique uniquement lorsque	Coût global le plus élevé si vous prévoyez une utilisation permanente. Toutefois, il s'agit du coût global le plus bas si vous prévoyez d'utiliser votre nœud réservé de manière peu fréquente, plus de 15 % environ d'une durée de trois ans.

Offre	Coût initial	Coût d'utilisation	Avantages
		vous exécutez le nœud réservé.	
Utilisation à la demande (aucun nœud réservé)	Aucun	Tarif horaire le plus élevé. Appliqué lorsque le nœud est en cours d'exécution.	Coût horaire le plus élevé.

Pour plus d'informations, consultez Amazon ElastiCache Pricing.

Obtenir des informations sur des offres de nœuds réservés

Avant d'acheter des nœuds réservés, vous pouvez obtenir des informations sur les offres disponibles.

Les exemples suivants montrent comment obtenir des prix et des informations sur les offres de nœuds réservés disponibles à l'aide de l' ElastiCache API AWS Management Console AWS CLI, et.

Rubriques

- Obtenir des informations sur des offres de nœuds réservés (console)
- Obtenir des informations sur des offres de nœuds réservés (AWS CLI)
- Obtenir des informations sur les offres de nœuds réservés (ElastiCache API)

Obtenir des informations sur des offres de nœuds réservés (console)

Pour obtenir les tarifs et d'autres informations sur les offres de clusters réservés disponibles à l'aide du AWS Management Console, suivez la procédure suivante.

Pour obtenir des informations sur les offres de nœuds réservées disponibles

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Nœuds réservés.
3. Choisissez Purchase Reserved Node (Acheter le nœud réservé).
4. Pour Engine, choisissez Valkey, Memcached ou Redis OSS.
5. Pour déterminer les offres disponibles, effectuez des sélections pour les trois options suivantes :

   - Type de nœud
   - Durée
   - Type d'offre

   Une fois que vous avez effectué ces sélections, le coût par nœud et le coût total de vos sélections s'affichent sous Reservation details (Détails de réservation).
6. Choisissez Annuler pour éviter d'acheter ces nœuds et d'avoir à payer des frais.

Obtenir des informations sur des offres de nœuds réservés (AWS CLI)

Pour obtenir les tarifs et d'autres informations sur les offres de nœuds réservés disponibles pour Valkey ou Redis OSS, tapez la commande suivante à l'invite de commande :

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.large",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "redis",
            "OfferingType": "All Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.X,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
 },
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.xlarge",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "redis",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXX,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
 },
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.large",
            "Duration": 31536000,
```

```
            "FixedPrice": X.X,
            "UsagePrice": X.X,
            "ProductDescription": "redis",
            "OfferingType": "No Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXX,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
}
```

Pour obtenir les tarifs et d'autres informations sur les offres de nœuds réservés disponibles pour Memcached, tapez la commande suivante à l'invite de commande :

```
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.large",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "All Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.X,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
    },
 {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xxx.xlarge",
            "Duration": 94608000,
            "FixedPrice": XXXX.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXXX,
                    "RecurringChargeFrequency": "Hourly"
                }
```

```
            ]
        },
        {
            "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
            "CacheNodeType": "cache.xx.12xlarge",
            "Duration": 31536000,
            "FixedPrice": X.X,
            "UsagePrice": X.X,
            "ProductDescription": "memcached",
            "OfferingType": "No Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": X.XXXX,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
}
```

Pour plus d'informations, voir describe-reserved-cache-nodes-offerings dans la AWS CLI référence.

Obtenir des informations sur les offres de nœuds réservés (ElastiCache API)

Pour obtenir une tarification et des informations sur les offres de nœuds réservés disponibles, appelez l'action `DescribeReservedCacheNodesOfferings`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReservedCacheNodesOfferings
    &Version=2014-12-01
    &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>
```

Pour plus d'informations, consultez DescribeReservedCacheNodesOfferingsla référence de ElastiCache l'API.

Achat d'un nœud réservé

Les exemples suivants montrent comment acheter une offre de nœuds réservés à l'aide de l'API AWS Management Console AWS CLI, du et de l' ElastiCache API.

> **Important**
>
> Si vous suivez les exemples présentés dans cette section, des frais sont débités sur votre AWS compte que vous ne pouvez pas annuler.

Rubriques

- Achat d'un nœud réservé (console)
- Achat d'un nœud réservé (AWS CLI)
- Achat d'un nœud réservé (ElastiCache API)

Achat d'un nœud réservé (console)

Cet exemple montre l'achat d'une offre spécifique de nœud de réservé, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, avec l'ID de nœud de réservé myreservationID.

La procédure suivante utilise le AWS Management Console pour acheter l'offre de nœuds réservés en proposant un identifiant.

Pour acheter des nœuds réservés

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste de navigation, choisissez le lien Reserved Nodes (Nœuds réservés).
3. Choisissez le bouton Purchase reserved nodes (Acheter les nœuds réservés).
4. Pour Engine, choisissez Valkey, Memcached ou Redis OSS.
5. Pour déterminer les offres disponibles, effectuez des sélections pour les trois options suivantes :
   - Type de nœud
   - Durée
   - Type d'offre
   - Un paramètre Reserved node ID (ID du nœud réservé) facultatif

Une fois que vous avez effectué ces sélections, le coût par nœud et le coût total de vos sélections s'affichent sous Reservation details (Détails de réservation).

6. Choisissez Purchase (Acheter).

Achat d'un nœud réservé (AWS CLI)

L'exemple suivant montre l'achat d'une offre spécifique de cluster réservé, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, avec l'ID de nœud réservé myreservationID.

A partir d'une invite de commande, entrez la commande suivante :

Pour Linux, macOS ou Unix :

```
aws elasticache purchase-reserved-cache-nodes-offering \
    --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
    --reserved-cache-node-id myreservationID
```

Pour Windows :

```
aws elasticache purchase-reserved-cache-nodes-offering ^
    --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
    --reserved-cache-node-id myreservationID
```

La commande renvoie un résultat semblable à ce qui suit :

```
RESERVATION  ReservationId      Class           Start Time                Duration
 Fixed Price  Usage Price  Count  State            Description      Offering Type
RESERVATION  myreservationid    cache.xx.small  2013-12-19T00:30:23.247Z  1y
 XXX.XX USD   X.XXX USD    1      payment-pending  memcached        Medium Utilization
```

Pour plus d'informations, voir purchase-reserved-cache-nodes-offering dans la AWS CLI référence.

Achat d'un nœud réservé (ElastiCache API)

L'exemple suivant montre l'achat d'une offre spécifique de nœud réservé, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, avec l'ID de cluster réservé myreservationID.

Appelez l'opération `PurchaseReservedCacheNodesOffering` avec les paramètres suivants :

- `ReservedCacheNodesOfferingId` = `649fd0c8-cf6d-47a0-bfa6-060f8e75e95f`
- `ReservedCacheNodeID` = `myreservationID`
- `CacheNodeCount` = `1`

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=PurchaseReservedCacheNodesOffering
    &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
    &ReservedCacheNodeID=myreservationID
    &CacheNodeCount=1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez PurchaseReservedCacheNodesOfferingla référence de ElastiCache l'API.

Obtenir des informations sur vos nœuds réservés

Vous pouvez obtenir des informations sur les nœuds réservés que vous avez achetés à l'aide du AWS Management Console AWS CLI, du et de l' ElastiCache API.

Rubriques

- Obtenir des informations sur vos nœuds réservés (console)
- Obtenir des informations sur vos nœuds réservés (AWS CLI)
- Obtenir des informations sur vos nœuds réservés (ElastiCache API)

Obtenir des informations sur vos nœuds réservés (console)

La procédure suivante décrit comment utiliser le pour AWS Management Console obtenir des informations sur les nœuds réservés que vous avez achetés.

Pour obtenir des informations sur vos nœuds réservées achetés

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste de navigation, choisissez le lien Reserved nodes (Nœuds réservés).

   Les nœuds réservés pour votre compte apparaissent dans la liste Nœuds réservés. Vous pouvez choisir n'importe quel nœud réservé dans la liste pour consulter des informations détaillées correspondantes dans le volet de détails au bas de la console.

Obtenir des informations sur vos nœuds réservés (AWS CLI)

Pour obtenir des informations sur les nœuds réservés à votre AWS compte, tapez la commande suivante à l'invite de commande :

```
aws elasticache describe-reserved-cache-nodes
```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
    "ReservedCacheNodeId": "myreservationid",
    "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
    "CacheNodeType": "cache.xx.small",
```

```
    "DataTiering": "disabled",
    "Duration": "31536000",
    "ProductDescription": "memcached",
    "OfferingType": "Medium Utilization",
    "MaxRecords": 0
}
```

Pour plus d'informations, reportez-vous describe-reserved-cache-nodesà la section AWS CLI Référence.

Obtenir des informations sur vos nœuds réservés (ElastiCache API)

Pour obtenir des informations sur les nœuds réservés à votre AWS compte, appelez l'`DescribeReservedCacheNodes`opération.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReservedCacheNodes
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez DescribeReservedCacheNodesla référence de ElastiCache l'API.

Migration des nœuds de génération précédente

Les nœuds de génération précédente sont des types de nœuds qui sont en cours d'élimination progressive. Si aucun cluster existant n'utilise un type de nœud de génération précédente, ElastiCache ne prend pas en charge la création de nouveaux clusters avec ce type de nœud.

En raison du nombre limité de types de nœuds de génération précédente, nous ne pouvons pas garantir un remplacement réussi lorsqu'un nœud devient défectueux dans votre ou vos clusters. Dans un tel scénario, la disponibilité de votre cluster peut être affectée négativement.

Nous vous recommandons de migrer votre ou vos clusters vers un nouveau type de nœud pour une disponibilité et des performances accrues. Pour obtenir un type de nœud recommandé vers lequel migrer, veuillez consulter Chemins de mise à niveau. Pour obtenir la liste complète des types de nœuds pris en charge et des types de nœuds de génération précédente dans ElastiCache, consultezTypes de nœuds pris en charge.

Migration de nœuds sur un cluster Valkey ou Redis OSS

La procédure suivante décrit comment migrer votre type de nœud de cluster Valkey ou Redis OSS à l'aide de la ElastiCache console. Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal. Selon la configuration de votre cluster, vous pouvez constater les temps d'arrêt suivants. Voici des estimations qui peuvent différer en fonction de vos configurations spécifiques :

- Le mode cluster désactivé (nœud unique) peut expérimenter environ 60 secondes de temps d'arrêt, principalement en raison de la propagation DNS.
- Le mode cluster désactivé (avec nœud de réplication) peut prendre environ 1 seconde pour les clusters exécutant Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.6 et versions ultérieures. Toutes les versions inférieures peuvent prendre environ 10 secondes.
- Le mode cluster activé peut expérimenter un temps d'arrêt d'environ 1 seconde.

Pour modifier un type de nœud de cluster Valkey ou Redis OSS à l'aide de la console :

1. Connectez-vous à la console et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Dans la liste des clusters, choisissez le cluster que vous souhaitez migrer.
4. Choisissez Actions, puis Modifier.
5. Choisissez le nouveau type de nœud dans la liste des types de nœuds.
6. Si vous souhaitez effectuer un processus de mise à l'échelle immédiatement, cochez la case Apply immediately (Appliquer immédiatement). Si la case Apply immediately (Appliquer immédiatement) est décochée, le processus de mise à l'échelle est effectué lors de la fenêtre de maintenance suivante du cluster.
7. Sélectionnez Modifier. Si vous avez choisi Apply immediately lors de l'étape précédente, le statut du cluster passe à modification. Lorsque son statut passe à available, la modification est terminée et vous pouvez commencer à utiliser le nouveau cluster.

Pour modifier un type de nœud de cluster Valkey ou Redis OSS à l'aide de : AWS CLI

Utilisez l'modify-replication-groupAPI comme indiqué ci-dessous :

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group /
 --replication-group-id my-replication-group /
 --cache-node-type new-node-type /
 --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
 --replication-group-id my-replication-group ^
 --cache-node-type new-node-type ^
 --apply-immediately
```

Dans ce scénario, la valeur de *new-node-type* est le type de nœud vers lequel vous effectuez la migration. En transmettant le paramètre `--apply-immediately`, la mise à jour sera appliquée immédiatement lorsque le groupe de réplication passe du statut modifying (en cours de modification) à available (disponible). Si la case Apply immediately (Appliquer immédiatement) est décochée, le processus de mise à l'échelle est effectué lors de la fenêtre de maintenance suivante du cluster.

> **Note**
>
> Si vous ne parvenez pas à modifier le cluster avec un`InvalidCacheClusterState`, vous devez d'abord supprimer un nœud ayant échoué à la restauration.

Fixation ou retrait restore-failed-node (s)

La procédure suivante décrit comment réparer ou supprimer un ou plusieurs nœuds dont la restauration a échoué de votre cluster Valkey ou Redis OSS. Pour en savoir plus sur la façon dont les ElastiCache nœuds entrent dans un état d'échec de restauration, consultez. Affichage de l'état du ElastiCache nœud Nous recommandons de supprimer d'abord tous les nœuds dont la restauration a échoué, puis de migrer les nœuds de génération précédente restants du ElastiCache cluster vers un type de nœud de nouvelle génération, et enfin de rajouter le nombre de nœuds requis.

Pour supprimer le nœud en échec de restauration (console) :

1. Connectez-vous à la console et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Dans la liste de clusters, choisissez le nom du cluster dans lequel vous souhaitez supprimer un nœud.
4. Dans la liste des partitions, choisissez la partition dont vous souhaitez supprimer un nœud. Ignorez cette étape si le mode cluster est désactivé pour le cluster.
5. Dans la liste des nœuds, choisissez le nœud avec un statut de `restore-failed`.
6. Choisissez Actions, puis Delete node (Supprimer le nœud).

Une fois que vous avez supprimé le ou les nœuds dont la restauration a échoué de votre ElastiCache cluster, vous pouvez désormais migrer vers un type de nouvelle génération. Pour plus d'informations, consultez la section Migration de nœuds sur un cluster Valkey ou Redis OSS ci-dessus.

Pour réajouter des nœuds à votre ElastiCache cluster, consultezAjouter des nœuds à un ElastiCache cluster.

Migration de nœuds sur un cluster Memcached

ElastiCache Pour effectuer la migration de Memcached vers un autre type de nœud, vous devez créer un nouveau cluster, toujours vide au départ, que votre application peut remplir.

Pour migrer votre type de nœud de cluster ElastiCache pour Memcached à l'aide de la ElastiCache console :

- Créez un nouveau cluster de avec le nouveau type de nœud. Pour de plus amples informations, veuillez consulter Création d'un cluster Memcached (console).
- Créez un nouveau cluster de cache avec le nouveau type de nœud. Pour plus d'informations, consultez Trouver les points de terminaison d'un cluster (console) (Memcached)
- Supprimez l'ancien cluster. Pour de plus amples informations, veuillez consulter Supprimer un cluster dans ElastiCache.

Gestion des clusters dans ElastiCache

Un cluster est un ensemble d'un ou de plusieurs nœuds de cache, qui exécutent tous une instance du logiciel de moteur Valkey, Memcached et Redis OSS. Lorsque vous créez un cluster, vous spécifiez le moteur et la version que tous les nœuds utiliseront.

Clusters Valkey et Redis OSS

Le schéma suivant illustre un cluster Valkey ou Redis OSS typique. Ces clusters peuvent contenir un seul nœud ou jusqu'à six nœuds au sein d'une partition (API/CLI : groupe de nœuds), un cluster Valkey ou Redis OSS à nœud unique (mode cluster désactivé) ne possède aucune partition, et un cluster Valkey ou Redis OSS (mode cluster désactivé) à plusieurs nœuds ne possède qu'une seule partition. Les clusters Valkey ou Redis OSS (mode cluster activé) peuvent contenir jusqu'à 500 partitions, vos données étant partitionnées entre les partitions. La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur est Valkey 7.2 ou version ultérieure ou Redis OSS 5.0.6 ou version ultérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux. Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Lorsque vous avez plusieurs nœuds dans une partition Valkey ou Redis OSS, l'un des nœuds est un read/write nœud principal. Tous les autres nœuds de la partition sont des réplicas en lecture seule.

Les clusters Valkey ou Redis OSS typiques se présentent comme suit.

Clusters Memcached

Les clusters Memcached classiques se présentent comme suit. Les clusters Memcached contiennent de 1 à 60 nœuds, sur lesquels vous partitionnez vos données horizontalement.

Opérations Elasticache pour Valkey, Memcached et Redis OSS

La plupart ElastiCache des opérations 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' ElastiCache API et AWS CLI les commandes. L'identifiant du cluster doit être unique pour ce client dans une AWS région.

ElastiCache prend en charge plusieurs versions de moteur. Sauf si vous avez des raisons spécifiques, nous vous recommandons d'utiliser la dernière version.

ElastiCache les clusters sont conçus pour être accessibles via une EC2 instance Amazon. Si vous lancez votre cluster dans un cloud privé virtuel (VPC) basé sur le service Amazon VPC, vous pouvez y accéder depuis l'extérieur d' AWS. Pour de plus amples informations, veuillez consulter Accès aux ElastiCache ressources depuis l'extérieur AWS.

Pour obtenir la liste des versions prises en chargeMoteurs et versions pris en charge, reportez-vous aux Versions du moteur Redis OSS prises en charge sections, etPris en charge ElastiCache pour les versions de Memcached.

Choix d'un type de réseau dans ElastiCache

ElastiCache prend en charge les versions 4 et 6 du protocole Internet (IPv4 et IPv6), ce qui vous permet de configurer votre cluster pour qu'il accepte :

- uniquement IPv4 des connexions,
- uniquement IPv6 des connexions,
- à la fois IPv4 et IPv6 connexions (double pile)

IPv6 est pris en charge pour les charges de travail utilisant Valkey 7.2 et versions ultérieures, ou Redis OSS 6.2 et versions ultérieures, sur toutes les instances créées sur le système Nitro. Il n'y a pas de frais supplémentaires pour y ElastiCache accéder IPv6.

> **Note**
>
> La migration des clusters créés avant la disponibilité de IPV6 /dual-stack n'est pas prise en charge. La commutation entre les types de réseaux sur les clusters nouvellement créés n'est pas non plus prise en charge.

IPv6 est pris en charge pour les charges de travail utilisant Memcached 1.6.6 et versions ultérieures sur toutes les instances basées sur le système Nitro. Il n'y a pas de frais supplémentaires pour y ElastiCache accéder IPv6.

Configuration de sous-réseaux pour le type de réseau

Si vous créez un cluster dans un Amazon VPC, vous devez spécifier un groupe de sous-réseaux. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds. ElastiCache les clusters nécessitent un sous-réseau à double pile auquel les deux IPv6 adresses sont attribuées pour fonctionner en mode double pile IPv4 et un sous-réseau « uniquement » pour fonctionner en tant que « IPv6 -only ». IPv6

Utilisation de la double pile

Lors de l'utilisation ElastiCache de Redis OSS en mode cluster activé, du point de vue de l'application, la connexion à tous les nœuds du cluster via le point de terminaison de configuration n'est pas différente de la connexion directe à un nœud de cache individuel. Pour ce faire, un client

sensible aux clusters doit s'engager dans un processus de découverte des clusters et demander les informations de configuration pour tous les nœuds. Le protocole de découverte de Redis ne prend en charge qu'une seule adresse IP par nœud.

Lorsque vous créez un cluster de cache avec ElastiCache pour Memcached et que vous choisissez Dual-Stack comme type de réseau, vous devez ensuite désigner un type de découverte IP, soit. IPv4 IPv6 ElastiCache définira par défaut le type de réseau et la découverte IP IPv6, mais cela peut être modifié. Si vous utilisez la découverte automatique, seules les adresses IP du type d'IP que vous avez choisi sont renvoyées au client Memcached. Pour de plus amples informations, veuillez consulter Identifiez automatiquement les nœuds de votre cluster (Memcached).

Pour maintenir la rétrocompatibilité avec tous les clients existants, la découverte d'adresses IP est introduite, qui vous permet de sélectionner le type d'adresse IP (c'est-à-dire IPv4 ou IPv6) à annoncer dans le protocole de découverte. Bien que cela limite la découverte automatique à un seul type d'adresse IP, la double pile reste avantageuse pour les charges de travail activées en mode cluster, car elle permet des migrations (ou des annulations) d'un type d'adresse IP de découverte IPv4 vers un type d'adresse IP de IPv6 découverte sans interruption de service.

Clusters à double pile ElastiCache compatibles TLS

Lorsque TLS est activé pour les ElastiCache clusters, les fonctions de découverte de clusters telles que `cluster slotscluster shards`, et `cluster nodes` avec Valkey ou Redis OSS et `config get cluster` avec Memcached renvoient des noms d'hôte au lieu de. IPs Les noms d'hôtes sont ensuite utilisés au lieu de se IPs connecter au ElastiCache cluster et d'effectuer une poignée de main TLS. Cela signifie que les clients ne seront pas affectés par le paramètre de découverte d'adresses IP. Pour les clusters prenant en charge TLS, le paramètre de découverte d'adresses IP n'a aucun effet sur le protocole IP préféré. Au lieu de cela, le protocole IP utilisé sera déterminé par le protocole IP que le client préfère lors de la résolution des noms d'hôtes DNS.

Pour obtenir des exemples de configuration d'une préférence de protocole IP lors de la résolution de noms d'hôtes DNS, consultez Clusters à double pile ElastiCache compatibles TLS.

Utilisation du AWS Management Console (Valkey et Redis OSS)

Lorsque vous créez un cluster à l'aide de AWS Management Console, sous Connectivité, choisissez un type de réseau, soit un type de réseau IPv4, IPv6soit une double pile. Si vous créez un cluster Valkey ou Redis OSS (mode cluster activé) et que vous choisissez la double pile, vous devez sélectionner un type d'IP de découverte, soit IPv6 . IPv4

Pour plus d'informations, consultez Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console) ou Création d'un système d'exploitation Valkey ou Redis (mode cluster désactivé) (console).

Lorsque vous créez un groupe de réplication à l'aide du AWS Management Console, choisissez un type de réseau, soit un type de réseau IPv4, IPv6soit une double pile. Si vous choisissez la double pile, vous devez sélectionner un type d'IP de découverte, IPv6 soit IPv4.

Pour plus d'informations, consultez Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro ou Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro.

Utilisation du AWS Management Console (Memcached)

Lorsque vous créez un cluster de cache à l'aide de AWS Management Console, sous Connectivité, choisissez un type de réseau, soit un type de réseau IPv4, IPv6soit une double pile. Si vous choisissez la double pile, vous devez sélectionner un type d'IP de découverte, IPv6 soit IPv4.

Pour de plus amples informations, veuillez consulter Création d'un cluster Memcached (console).

Utilisation de la CLI avec Valkey, Memcached ou Redis OSS.

Système d'exploitation Redis

Lorsque vous créez un cluster de cache avec Valkey ou Redis OSS à l'aide de la CLI, vous utilisez la create-cache-clustercommande et spécifiez les paramètres `NetworkType` et `IPDiscovery` :

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine redis \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1 \
    --network-type dual_stack \
    --ip-discovery ipv4
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
    --cache-cluster-id "cluster-test" ^
```

```
    --engine redis ^
    --cache-node-type cache.m5.large ^
    --num-cache-nodes 1 ^
    --network-type dual_stack ^
    --ip-discovery ipv4
```

Lorsque vous créez un groupe de réplication avec le mode cluster désactivé à l'aide de la CLI, vous utilisez la create-replication-groupcommande et spécifiez les `IPDiscovery` paramètres `NetworkType` et :

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "demo cluster with replicas" \
   --num-cache-clusters 3 \
   --primary-cluster-id redis01 \
   --network-type dual_stack \
   --ip-discovery ipv4
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "demo cluster with replicas" ^
   --num-cache-clusters 3 ^
   --primary-cluster-id redis01 ^
   --network-type dual_stack ^
   --ip-discovery ipv4
```

Lorsque vous créez un groupe de réplication avec le mode cluster activé et que vous l'utilisez IPv4 pour la découverte d'adresses IP à l'aide de la CLI, vous utilisez la create-replication-groupcommande et spécifiez les `IPDiscovery` paramètres `NetworkType` et :

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
```

```
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv4 \
  --region us-east-1
```

Pour Windows :

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

Lorsque vous créez un groupe de réplication avec le mode cluster activé et que vous l'utilisez IPv6 pour la découverte d'adresses IP à l'aide de la CLI, vous utilisez la create-replication-groupcommande et spécifiez les `IPDiscovery` paramètres `NetworkType` et :

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv6 \
  --region us-east-1
```

Pour Windows :

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
```

```
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv6 ^
  --region us-east-1
```

Memcached

Lorsque vous créez un cluster de cache avec Memcached à l'aide de la CLI, vous utilisez la create-cache-clustercommande et spécifiez les paramètres `NetworkType` et `IPDiscovery` :

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine memcached \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1 \
    --network-type dual_stack \
    --ip-discovery ipv4
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
    --cache-cluster-id "cluster-test" ^
    --engine memcached ^
    --cache-node-type cache.m5.large ^
    --num-cache-nodes 1 ^
    --network-type dual_stack ^
    --ip-discovery ipv4
```

Identifiez automatiquement les nœuds de votre cluster (Memcached)

Pour les clusters exécutant le moteur Memcached, ElastiCache prend en charge la détection automatique, c'est-à-dire la capacité des programmes clients à identifier automatiquement tous les nœuds d'un cluster de cache, ainsi qu'à établir et à maintenir des connexions avec tous ces nœuds.

> **Note**
>
> La détection automatique est ajoutée pour les clusters de cache exécutés sur Amazon ElastiCache Memcached. Auto Discovery n'est pas disponible pour les moteurs Valkey ou Redis OSS.

Avec la découverte automatique, votre application n'a pas besoin d'être manuellement connectée à chaque nœud de cache. Votre application se connecte plutôt à un nœud Memcached et accède à la liste de tous les nœuds. Grâce à cette liste, votre application connaît l'ensemble des nœuds du cluster et peut se connecter à chacun d'entre eux. Vous n'avez pas besoin de définir de façon spécifique chaque point de terminaison d'un nœud de cache dans votre application.

Si vous utilisez un type de réseau à double pile sur votre cluster, Auto Discovery renverra uniquement IPv6 les adresses IPv4 ou, selon le type que vous sélectionnez. Pour plus d'informations, consultez Choix d'un type de réseau dans ElastiCache.

Tous les nœuds de cache dans le cluster conservent une liste de métadonnées sur tous les autres nœuds. Ces métadonnées sont mises à jour chaque fois que des nœuds sont ajoutés ou supprimés du cluster.

Rubriques

- Avantages de la découverte automatique avec Memcached
- Comment fonctionne la découverte automatique
- Utilisation de la découverte automatique
- Connexion manuelle aux nœuds de cache Memcached
- Ajouter Auto Discovery à votre bibliothèque cliente Memcached
- ElastiCache clients dotés d'une fonction de détection automatique

Avantages de la découverte automatique avec Memcached

Lorsque vous utilisez Memcached, Auto Discovery offre les avantages suivants :

- Lorsque vous augmentez le nombre de nœuds dans un cluster de cache, les nouveaux nœuds s'enregistrent automatiquement dans la configuration du point de terminaison et auprès de tous les autres nœuds. Quand vous supprimez des nœuds dans un cluster de cache, les nœuds supprimés se désenregistrent automatiquement. Dans les deux cas, tous les autres nœuds du cluster sont mis à jour avec les métadonnées les plus récentes du nœud de cache.
- Les défaillances de nœuds de cache sont détectées automatiquement ; les nœuds défaillants sont automatiquement remplacés.

> **Note**
>
> Jusqu'à ce que le nœud soit complètement remplacé, celui-ci continuera à être défaillant.

- Un programme client doit uniquement se connecter au point de terminaison de configuration. Passé ce délai, la bibliothèque de découverte automatique se connecte à tous les autres nœuds du cluster.
- Les programmes clients interrogent le cluster une fois par minute (si nécessaire, cet intervalle peut être ajusté). En cas de modifications de la configuration du cluster, telles que des nœuds ajoutés ou supprimés, le client reçoit une liste mise à jour des métadonnées. Puis le client se connecte à ces nœuds ou s'en déconnecte, selon le cas.

La découverte automatique est activée sur tous les clusters de cache ElastiCache Memcached. Vous n'avez pas besoin de redémarrer un de vos nœuds de cache pour utiliser cette fonctionnalité.

Comment fonctionne la découverte automatique

Rubriques

- Connexion aux nœuds de cache
- Opérations de routine sur un cluster
- Autres opérations

Cette section décrit comment les applications clientes utilisent le client de ElastiCache cluster pour gérer les connexions aux nœuds de cache et interagir avec les éléments de données du cache.

Connexion aux nœuds de cache

Pour une application, il n'y a pas de différence entre se connecter au point de terminaison de la configuration du cluster et se connecter directement à chaque nœud de cache. Le graphique ci-dessous illustre le processus de connexion aux nœuds de cache.

Processus de connexion aux nœuds de cache

- Cette application résout le nom DNS du point de terminaison de la configuration. Etant donné que le point de terminaison de la configuration conserve les entrées CNAME pour tous les nœuds de cache, le nom DNS est converti en l'un de ces nœuds ; le client peut alors se connecter à ce nœud.
- Le client requiert les informations de configuration pour tous les autres nœuds. Du fait que chaque nœud conserve les informations de configuration de tous les nœuds du cluster, chaque nœud peut transmettre les informations de configuration au client à la demande.

- Le client reçoit la liste actuelle des adresses IP et des noms d'hôtes des nœuds de cache. Il peut alors se connecter à tous les autres nœuds du cluster.

> **Note**
>
> Le programme client actualise sa liste des adresses IP et de noms d'hôtes des nœuds de cache chaque minute. Cet intervalle peut être modifié si nécessaire.

Opérations de routine sur un cluster

Lorsque l'application est connectée à tous les nœuds de cache, le client de ElastiCache cluster détermine quels nœuds doivent stocker des éléments de données individuels et quels nœuds doivent être interrogés ultérieurement pour ces éléments de données. Le diagramme, ci-dessous, illustre le déroulement normal des opérations sur un cluster.

Déroulement des opérations de routine sur un cluster

- L'application émet une requête get pour une donnée particulière, identifiée par sa clé.
- Le client utilise un algorithme de hachage sur la clé pour déterminer quel nœud de cache contient l'élément de donnée.
- L'élément de données est demandé au nœud approprié.
- L'élément de données est transmis à l'application.

Autres opérations

Dans certains cas, vous pouvez modifier les nœuds d'un cluster. Par exemple, vous pouvez ajouter un nœud supplémentaire pour répondre à une demande supplémentaire ou supprimer un nœud pour économiser de l'argent pendant les périodes de demande réduite. Ou vous pouvez remplacer un nœud en raison d'une défaillance d'une sorte ou d'une autre.

Lorsque survient une modification dans un cluster qui requiert une mise à jour des métadonnées aux points de terminaison du cluster, cette modification affecte tous les nœuds en même temps. Ainsi, les métadonnées dans un nœud quelconque sont identiques à toutes les métadonnées de tous les autres nœuds du cluster.

Dans tous les cas, les métadonnées sont identiques dans tous les nœuds à chaque instant, puisque celles-ci sont mises à jour au même moment pour tous les nœuds dans le cluster. Vous devez toujours utiliser le point de terminaison de configuration pour obtenir les points de terminaison de chacun des nœuds du cluster. L'utilisation du point de terminaison de configuration vous évite d'obtenir des données de point de terminaison d'un nœud qui « disparaîtra » au moment où vous voulez y accéder.

Ajout d'un nœud

Pendant la phase d'activation d'un nœud, son point de terminaison n'est pas inclus dans les métadonnées. Dès que le nœud devient disponible, il est ajouté aux métadonnées de chacun des nœuds du cluster. Dans ce scénario, les métadonnées sont identiques dans tous les nœuds et vous pourrez interagir avec le nouveau nœud uniquement une fois qu'il est disponible. Avant que le nœud ne soit disponible, il ne vous apparaîtra pas et vous utiliserez les nœuds de votre cluster comme si ce nouveau nœud n'existait pas.

Suppression d'un nœud

Lorsqu'un nœud est supprimé, son point de terminaison est d'abord supprimé des métadonnées, puis le nœud est supprimé du cluster. Dans ce scénario, les métadonnées sont identiques dans tous les nœuds et à aucun moment elles ne contiendront le point de terminaison du nœud à supprimer alors que celui-ci est indisponible. Pendant la suppression du nœud, il n'est pas inclus dans les métadonnées et votre application interagira donc uniquement avec les n-1 nœuds restants, comme si ce nœud n'existait pas.

Remplacement d'un nœud

Si un nœud tombe en panne, ElastiCache démonte ce nœud et lance un nœud de remplacement. Le processus de remplacement prend quelques minutes. Pendant ce temps, les métadonnées dans

tous les nœuds contiennent encore le point de terminaison du nœud défaillant, mais toute tentative d'interaction avec celui-ci échouera. Par conséquent, vous devez toujours penser à réessayer.

Utilisation de la découverte automatique

Pour commencer à utiliser Auto Discovery avec ElastiCache for Memcached, procédez comme suit :

- Obtenir le point de terminaison de configuration
- Téléchargez le client ElastiCache de cluster
- Modifiez votre programme de candidature

Obtenir le point de terminaison de configuration

Pour se connecter à un cluster, les programmes clients doivent connaître le point de terminaison de configuration du cluster. Consultez la rubrique Trouver les points de terminaison d'un cluster (console) (Memcached)

Vous pouvez également utiliser la commande `aws elasticache describe-cache-clusters` avec le paramètre `--show-cache-node-info` :

Quelle que soit la méthode que vous utilisez pour trouver les points de terminaison du cluster, le point de terminaison de configuration aura toujours .cfg dans son adresse.

Example Recherche de points de terminaison à l'aide du for AWS CLI ElastiCache

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id mycluster \
    --show-cache-node-info
```

Pour Windows :

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id mycluster ^
    --show-cache-node-info
```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
    "CacheClusters": [
        {
            "Engine": "memcached",
            "CacheNodes": [
```

```
                {
                    "CacheNodeId": "0001",
                    "Endpoint": {
                        "Port": 11211,
                        "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
                    },
                    "CacheNodeStatus": "available",
                    "ParameterGroupStatus": "in-sync",
                    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
                    "CustomerAvailabilityZone": "us-east-1e"
                },
                {
                    "CacheNodeId": "0002",
                    "Endpoint": {
                        "Port": 11211,
                        "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
                    },
                    "CacheNodeStatus": "available",
                    "ParameterGroupStatus": "in-sync",
                    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
                    "CustomerAvailabilityZone": "us-east-1a"
                }
            ],
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "CacheParameterGroupName": "default.memcached1.4",
                "ParameterApplyStatus": "in-sync"
            },
            "CacheClusterId": "mycluster",
            "PreferredAvailabilityZone": "Multiple",
            "ConfigurationEndpoint": {
                "Port": 11211,
                "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
            },
            "CacheSecurityGroups": [],
            "CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterStatus": "available",
            "NumCacheNodes": 2,
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "CacheSubnetGroupName": "default",
            "EngineVersion": "1.4.24",
            "PendingModifiedValues": {},
```

```
            "PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
            "CacheNodeType": "cache.r3.large"
        }
    ]
}
```

Téléchargez le client ElastiCache de cluster

Pour pouvoir bénéficier de la découverte automatique, les programmes clients doivent utiliser le client de cluster ElastiCache. Le client de ElastiCache cluster est disponible pour Java, PHP et .NET et contient toute la logique nécessaire pour découvrir et se connecter à tous vos nœuds de cache.

Pour télécharger le client ElastiCache de cluster

1. Connectez-vous à la console AWS de gestion et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans la ElastiCache console, choisissez ElastiCache Cluster Client, puis Download.

Le code source du client ElastiCache Cluster pour Java est disponible à l'adresse https://github.com/amazonwebservices/aws-elasticache-cluster-client- memcached-for-java. Cette bibliothèque est basée sur le client Spymemcached populaire. Le client de ElastiCache cluster est publié sous la licence logicielle Amazon https://aws.amazon.com/asl. Vous êtes libre de modifier le code source selon vos besoins. Vous pouvez même incorporer le code dans d'autres bibliothèques Memcached open source, ou dans votre propre code client.

> **Note**
>
> Pour utiliser le client de ElastiCache cluster pour PHP, vous devez d'abord l'installer sur votre EC2 instance Amazon. Pour de plus amples informations, veuillez consulter Installation du client de ElastiCache cluster pour PHP.
> Pour un client qui prend en charge TLS, téléchargez le binaire avec PHP version 7.4 ou supérieure.
> Pour utiliser le client de ElastiCache cluster pour .NET, vous devez d'abord l'installer sur votre EC2 instance Amazon. Pour de plus amples informations, veuillez consulter Installation du client de ElastiCache cluster pour .NET.

Modifiez votre programme de candidature

Modifiez votre programme d'application afin qu'il utilise la découverte automatique. Les sections suivantes montrent comment utiliser le client de ElastiCache cluster pour Java, PHP et .NET.

> **Important**
>
> Lorsque vous spécifiez le point de terminaison de configuration du cluster, assurez-vous que le point de terminaison a « .cfg » dans son adresse comme illustré ici. N'utilisez pas un CNAME ou un point de terminaison sans « .cfg ».
>
> ```
> "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
> ```
>
> Si vous ne spécifiez pas explicitement le point de terminaison de configuration du cluster, la configuration se fait sur un nœud spécifique.

Utilisation du client de ElastiCache cluster pour Java

Le programme ci-dessous montre comment utiliser le client de ElastiCache cluster pour se connecter à un point de terminaison de configuration de cluster et ajouter un élément de données au cache. L'utilisation de la découverte automatique permet au programme de connecter tous les nœuds du cluster sans aucune autre intervention.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
                                 new InetSocketAddress(configEndpoint,
```

```
                                                        clusterPort));
        // The client will connect to the other cache nodes automatically.

        // Store a data item for an hour.
        // The client will decide which cache host will store this item.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Utilisation du client de ElastiCache cluster pour PHP

Le programme ci-dessous montre comment utiliser le client de ElastiCache cluster pour se connecter à un point de terminaison de configuration de cluster et ajouter un élément de données au cache. L'utilisation de la découverte automatique permettra au programme de se connecter à tous les nœuds du cluster sans aucune autre intervention.

Pour utiliser le client de ElastiCache cluster pour PHP, vous devez d'abord l'installer sur votre EC2 instance Amazon. Pour de plus amples informations, consultez Installation du client de ElastiCache cluster pour PHP.

```
<?php

 /**
  * Sample PHP code to show how to integrate with the Amazon ElastiCache
  * Auto Discovery feature.
  */

  /* Configuration endpoint to use to initialize memcached client.
   * This is only an example.  */
  $server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

  /* Port for connecting to the ElastiCache cluster.
   * This is only an example  */
  $server_port = 11211;

 /**
  * The following will initialize a Memcached client to utilize the Auto Discovery
 feature.
  *
  * By configuring the client with the Dynamic client mode with single endpoint, the
  * client will periodically use the configuration endpoint to retrieve the current
 cache
```

```
 * cluster configuration. This allows scaling the cache cluster up or down in number
of nodes
 * without requiring any changes to the PHP application.
 *
 * By default the Memcached instances are destroyed at the end of the request.
 * To create an instance that persists between requests,
 *    use persistent_id to specify a unique ID for the instance.
 * All instances created with the same persistent_id will share the same connection.
 * See http://php.net/manual/en/memcached.construct.php for more information.
 */
 $dynamic_client = new Memcached('persistent-id');
 $dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
 $dynamic_client->addServer($server_endpoint, $server_port);

 /**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
 $dynamic_client->set('key', 'value', 60);


/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
 $static_client = new Memcached('persistent-id');
 $static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
 $static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
 $static_client->set('key', 'value');
 ?>
```

Pour un exemple d'utilisation du client de ElastiCache cluster avec TLS activé, voir Utilisation du chiffrement en transit avec PHP et Memcached.

Utilisation du client de ElastiCache cluster pour .NET

> **Note**
>
> Le client de cluster ElastiCache .NET est devenu obsolète en mai 2022.

Le client .NET pour ElastiCache est open source à l'adresse https://github.com/awslabs/elasticache-cluster-config-net.

Les applications .NET obtiennent généralement leur configuration à partir de leur fichier de configuration. Le fichier ci-dessous est un exemple de fichier de configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <configSections>
        <section
            name="clusterclient"
            type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
 Amazon.ElastiCacheCluster" />
    </configSections>

    <clusterclient>
        <!-- the hostname and port values are from step 1 above -->
        <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
 port="11211" />
    </clusterclient>
</configuration>
```

Le programme C# ci-dessous montre comment utiliser le client de ElastiCache cluster pour se connecter à un point de terminaison de configuration de cluster et ajouter un élément de données au cache. L'utilisation de la découverte automatique permettra au programme de se connecter à tous les nœuds du cluster sans aucune autre intervention.

```
// *****************
// Sample C# code to show how to integrate with the Amazon ElastiCcache Auto Discovery
 feature.

using System;

using Amazon.ElastiCacheCluster;
```

```
using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo  {

    public static void Main(String[] args)  {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    }  // end Main

}  // end class DotNetAutoDiscoverDemo
```

Connexion manuelle aux nœuds de cache Memcached

Si votre programme client n'utilise pas Auto Discovery, il peut se connecter manuellement à chacun des nœuds de cache Memcached. C'est le comportement par défaut des clients Memcached.

Vous pouvez obtenir une liste des noms d'hôte des numéros de port des nœuds de cache à partir de la console de gestion AWS. Vous pouvez également utiliser la AWS CLI `aws elasticache describe-cache-clusters` commande avec le `--show-cache-node-info` paramètre.

Example

L'extrait du code Java, ci-dessous, illustre comment se connecter à tous les nœuds dans un cluster de cache contenant 4 nœuds :

```
...

ArrayList<String> cacheNodes = new ArrayList<String>(
 Arrays.asList(
     "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",
     "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));

MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));

...
```

> **Important**
>
> Si vous redimensionnez votre cluster de cache en ajoutant ou en supprimant des nœuds, vous devrez mettre à jour la liste des nœuds dans le code client.

Ajouter Auto Discovery à votre bibliothèque cliente Memcached

Les informations de configuration pour Auto Discovery sont stockées de manière redondante dans chaque nœud de cluster de cache Memcached. Les applications client peuvent interroger chaque nœud de cache pour obtenir les informations de configuration de tous les nœuds du cluster.

La façon de laquelle une application effectue cette tâche dépend de la version du moteur de cache :

- Si la version du moteur de cache est 1.4.14 ou supérieure, utilisez la commande `config`.
- Si la version du moteur de cache est inférieure à 1.4.14, utilisez la commande `get AmazonElastiCache:cluster`.

Les résultats de ces deux commandes sont identiques et sont décrits dans la section Résultat ci-dessous.

Moteur de cache version 1.4.14 ou supérieure

Dans le cas d'une version 1.4.14 ou supérieure du moteur de cache, utilisez la commande `config`. Cette commande a été ajoutée aux protocoles ASCII et binaires Memcached par ElastiCache et est implémentée dans le client de cluster. ElastiCache Si vous souhaitez utiliser la découverte automatique avec une autre bibliothèque client, alors cette bibliothèque devra être étendue pour prendre en charge la commande `config`.

> **Note**
>
> La documentation suivante concerne le protocole ASCII ; toutefois, la commande `config` prend en charge le protocole ASCII aussi bien que le protocole binaire. Si vous souhaitez ajouter la prise en charge de la découverte automatique à l'aide du protocole binaire, reportez-vous au code source du client de ElastiCache cluster.

Syntaxe

```
config [sub-command] [key]
```

Options

Name (Nom)	Description	Obligatoire
`sub-command`	La sous-commande utilisée pour interagir avec un nœud de cache. Pour la découverte automatique, cette sous-commande est `get`.	Oui
`key`	La clé sous laquelle la configuration du cluster est sauvegardée. Pour la découverte automatique, cette clé est appelée `cluster`.	Oui

Pour obtenir les informations de configuration du cluster, utilisez la commande suivante :

```
config get cluster
```

Moteur de cache version 1.4.14 ou inférieure

Pour obtenir les informations de configuration du cluster, utilisez la commande suivante :

```
get AmazonElastiCache:cluster
```

> **Note**
>
> Ne modifiez pas la clé « :cluster AmazonElastiCache », car c'est là que se trouvent les informations de configuration du cluster. Si vous remplacez cette clé, il se peut que le client soit mal configuré pendant une courte période (pas plus de 15 secondes) avant de mettre à jour ElastiCache automatiquement et correctement les informations de configuration.

Résultat

Que vous utilisiez `config get cluster` ou `get AmazonElastiCache:cluster`, la réponse tient deux lignes :

- Le numéro de version des informations de configuration. Chaque fois qu'un nœud est ajouté ou supprimé du cluster de cache, le numéro de la version s'incrémente d'une unité.

- Une liste des nœuds de cache. Chaque nœud dans la liste est représenté par un groupe hostname|ip-address|port et chaque nœud est séparé par un espace.

Un retour à la ligne et un saut de ligne (CR + LF) figurent à la fin de chaque ligne. La ligne de données se termine par un retour à la ligne, puis un retour à la ligne et un saut de ligne (CR + LF) sont ajoutés. La ligne où figure la version de configuration se termine par saut de ligne (LF) sans retour à la ligne (CR).

Un cluster de cache contenant trois nœuds est représenté comme suit :

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Pour chaque nœud, sont affichés le CNAME et l'adresse IP privée. Le CNAME est toujours affiché ; l'adresse IP n'est affichée que si elle existe. Cependant, les barres verticales « `|` »seront tout de même affichées.

Example

Voici un exemple de la charge utile transmise lorsque vous interrogez les informations de configuration :

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
 myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

> **Note**
>
> - La deuxième ligne indique que les informations de configuration a été modifiées douze fois jusqu'à présent.
> - Sur la troisième ligne, la liste de nœuds est triée alphabétiquement par nom d'hôte. Cet ordre peut être différent de celui que vous utilisez habituellement dans votre application client.

ElastiCache clients dotés d'une fonction de détection automatique

Les programmes clients de cluster peuvent identifier et se connecter automatiquement à tous les nœuds de cluster de cache qui exécutent le moteur Memcached.

Cette section décrit l'installation et la configuration des clients ElastiCache PHP et .NET pour une utilisation avec la découverte automatique.

Rubriques

- Installation et compilation des clients de cluster
- Configuration des ElastiCache clients

Installation et compilation des clients de cluster

Cette section traite de l'installation, de la configuration et de la compilation des clients du cluster de découverte ElastiCache automatique Amazon PHP et .NET.

Rubriques

- Installation du client de ElastiCache cluster pour .NET
- Installation du client de ElastiCache cluster pour PHP
- Compiler le code source du client de ElastiCache cluster pour PHP

Installation du client de ElastiCache cluster pour .NET

Vous pouvez trouver le code du client ElastiCache .NET Cluster en open source à l'adresse https://github.com/awslabs/elasticache-cluster-config-net.

Cette section décrit comment installer, mettre à jour et supprimer les composants .NET pour le client de ElastiCache cluster sur les EC2 instances Amazon. Pour plus d'informations sur la découverte automatique, consultez Identifiez automatiquement les nœuds de votre cluster (Memcached). Pour un exemple de code .NET pour utiliser le client, consultez Utilisation du client de ElastiCache cluster pour .NET.

Rubriques

- Installation de .NET
- Téléchargez le client de cluster ElastiCache .NET pour ElastiCache

- Installez les AWS assemblages avec NuGet

Installation de .NET

.NET 3.5 ou version ultérieure doit être installé pour utiliser le SDK AWS .NET pour ElastiCache. Si vous ne disposez pas .NET 3.5 ou version ultérieure, vous pouvez télécharger et installer la dernière version de http://www.microsoft.com/net.

Téléchargez le client de cluster ElastiCache .NET pour ElastiCache

Pour télécharger le client de cluster ElastiCache .NET

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, cliquez sur ElastiCache Cluster Client.
3. Dans la liste Télécharger le client de cluster ElastiCache Memcached, sélectionnez .NET, puis cliquez sur Télécharger.

Installez les AWS assemblages avec NuGet

NuGet est un système de gestion de paquets pour la plate-forme .NET. NuGet connaît les dépendances de l'assemblage et installe automatiquement tous les fichiers requis. NuGet les assemblages installés sont stockés avec votre solution, plutôt que dans un emplacement central, par exemple`Program Files`, afin que vous puissiez installer des versions spécifiques à une application sans créer de problèmes de compatibilité.

Installation NuGet

NuGet peut être installé à partir de la galerie d'installation sur MSDN ; voir https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c. Si vous utilisez Visual Studio 2010 ou version ultérieure, NuGet il est automatiquement installé.

Vous pouvez l'utiliser NuGet à partir de l'Explorateur de solutions ou de la console Package Manager.

Utilisation NuGet depuis l'Explorateur de solutions

À utiliser NuGet depuis l'Explorateur de solutions dans Visual Studio 2010

1. Dans le menu Tools, sélectionnez Library Package Manager.

2. Cliquez sur Console du Gestionnaire de package.

À utiliser NuGet depuis l'Explorateur de solutions dans Visual Studio 2012 ou Visual Studio 2013

1. Dans le menu Outils, sélectionnez NuGet Package Manager.
2. Cliquez sur Console du Gestionnaire de package.

A partir de la ligne de commande, vous pouvez installer les assemblages à l'aide de `Install-Package`, comme indiqué ci-après.

```
Install-Package Amazon.ElastiCacheCluster
```

Pour voir une page pour chaque package disponible NuGet, tel que les assemblages AWS SDK et AWS.Extensions, consultez le NuGet site Web à l'adresse http://www.nuget.org. La page de chaque package inclut un exemple de ligne de commande pour installer le package à l'aide de la console et une liste des versions précédentes du package disponibles viaNuGet.

Pour plus d'informations sur les commandes de Console du Gestionnaire de package, accédez à http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29.

Installation du client de ElastiCache cluster pour PHP

Cette section décrit comment installer, mettre à jour et supprimer les composants PHP pour le client de ElastiCache cluster sur les EC2 instances Amazon. Pour plus d'informations sur la découverte automatique, consultez Identifiez automatiquement les nœuds de votre cluster (Memcached). Pour un exemple de code PHP pour utiliser le client, consultez Utilisation du client de ElastiCache cluster pour PHP.

Rubriques

- Téléchargement du package d'installation
- Pour les utilisateurs qui ont déjà l'extension php-memcached installée
- Étapes d'installation pour les nouveaux utilisateurs
- Suppression du client de cluster PHP

Téléchargement du package d'installation

Pour vous assurer que vous utilisez la bonne version du client de ElastiCache cluster pour PHP, vous devez savoir quelle version de PHP est installée sur votre EC2 instance Amazon. Vous devez également savoir si votre EC2 instance Amazon exécute une version 64 bits ou 32 bits de Linux.

Pour déterminer la version de PHP installée sur votre EC2 instance Amazon

- A l'invite de commande, exécutez la commande suivante :

  ```
  php -v
  ```

  La version PHP s'affiche dans le résultat, comme dans cet exemple :

  ```
  PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
  Copyright (c) 1997-2012 The PHP Group
  Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
  ```

  > **Note**
  >
  > Si vos versions PHP et Memcached sont incompatibles, vous verrez un message d'erreur s'afficher, similaire à ce qui suit :
  >
  > ```
  > PHP Warning: PHP Startup: memcached: Unable to initialize module
  > ```

```
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

Si cela se produit, vous devrez compiler le module du code source. Pour de plus amples informations, veuillez consulter Compiler le code source du client de ElastiCache cluster pour PHP.

Pour déterminer votre architecture Amazon EC2 AMI (64 bits ou 32 bits)

1. Connectez-vous à la EC2 console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans la liste des instances, cliquez sur votre EC2 instance Amazon.
3. Dans l'onglet Description, recherchez le champ AMI. Une instance 64 bits doit disposer `x86_64` dans la description ; pour une instance 32 bits, recherchez `i386` ou `i686` dans ce champ.

Vous êtes maintenant prêt à télécharger le client de ElastiCache cluster.

Pour télécharger le client de ElastiCache cluster pour PHP

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la ElastiCache console, choisissez ElastiCache Cluster Client.
3. Dans la liste Télécharger le client de cluster ElastiCache Memcached, choisissez le client de ElastiCache cluster qui correspond à votre version de PHP et à votre architecture AMI, puis cliquez sur le bouton Télécharger.

Pour les utilisateurs qui ont déjà l'extension php-memcached installée

Pour mettre à jour l'installation **`php-memcached`**

1. Supprimez l'installation précédente de l'extension de Memcached pour PHP, tel que décrit par la rubrique Suppression du client de cluster PHP.
2. Installez la nouvelle ElastiCache `php-memcached` extension comme décrit précédemment dansÉtapes d'installation pour les nouveaux utilisateurs.

Étapes d'installation pour les nouveaux utilisateurs

Rubriques

- Installation de PHP 7.x pour les nouveaux utilisateurs
- Installation de PHP 5.x pour les nouveaux utilisateurs

Installation de PHP 7.x pour les nouveaux utilisateurs

Rubriques

- Pour installer PHP 7 sur un serveur Ubuntu 14.04 LTS AMI (64 bits et 32 bits)
- Pour installer PHP 7 sur une AMI Amazon Linux 201609
- Pour installer PHP 7 sur une AMI de SUSE Linux

Pour installer PHP 7 sur un serveur Ubuntu 14.04 LTS AMI (64 bits et 32 bits)

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez les commandes suivantes :

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Installez PHP 7.

```
sudo yum install php70
```

4. Téléchargez le client Amazon ElastiCache Cluster.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extrayez `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Avec les autorisations racine, copiez le fichier artefact extrait `amazon-elasticache-cluster-client.so` dans `/usr/lib/php/20151012`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Insérez la ligne `extension=amazon-elasticache-cluster-client.so` dans le fichier `/etc/php/7.0/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php/7.0/cli/php.ini
```

8. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 7 sur une AMI Amazon Linux 201609

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez la commande suivante :

```
sudo yum install gcc-c++
```

3. Installez PHP 7.

```
sudo yum install php70
```

4. Téléchargez le client Amazon ElastiCache Cluster.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extrayez `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Avec l'autorisation racine, copiez le fichier artefact extrait `amazon-elasticache-cluster-client.so` dans `/usr/lib64/php/7.0/modules/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Créez le fichier `50-memcached.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php-7.0.d/50-memcached.ini
```

8. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 7 sur une AMI de SUSE Linux

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez la commande suivante :

```
sudo zypper install gcc
```

3. Installez PHP 7.

```
sudo yum install php70
```

4. Téléchargez le client Amazon ElastiCache Cluster.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extrayez `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Avec l'autorisation racine, copiez le fichier artefact extrait `amazon-elasticache-cluster-client.so` dans `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Insérez la ligne `extension=amazon-elasticache-cluster-client.so` dans le fichier `/etc/php7/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php7/cli/php.ini
```

8. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Installation de PHP 5.x pour les nouveaux utilisateurs

Rubriques

- Pour installer PHP 5 sur une AMI d'Amazon Linux 2014.03 (64 bits et 32 bits)
- Pour installer PHP 5 sur une AMI de Red Hat Enterprise Linux 7,0 (64 bits et 32 bits)
- Pour installer PHP 5 sur une AMI de serveur Ubuntu 14.04 LTS (64 bits et 32 bits)
- Pour installer PHP 5 pour AMI de SUSE Linux Enterprise Server 11 (64 bits ou 32 bits)
- Autres distributions Linux

Pour installer PHP 5 sur une AMI d'Amazon Linux 2014.03 (64 bits et 32 bits)

1. Lancez une instance Amazon Linux (32 bits ou 64 bits) et connectez-vous.
2. Installez les dépendances PHP :

```
sudo yum install gcc-c++ php php-pear
```

3. Téléchargez le `php-memcached` package correspondant à votre EC2 instance Amazon et à votre version de PHP. Pour de plus amples informations, veuillez consulter Téléchargement du package d'installation.
4. Installer `php-memcached`. L'URI doit être le chemin de téléchargement du package d'installation :

```
sudo pecl install <package download path>
```

Voici un exemple de commande installation pour PHP 5.4, Linux 64 bits. Dans cet exemple, remplacez *X.Y.Z* par le numéro de version réel :

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

> **Note**
>
> Veuillez à utiliser la dernière version de l'artefact d'installation.

5. Avec root/sudo autorisation, ajoutez un nouveau fichier nommé `memcached.ini` dans le `/etc/php.d` répertoire et insérez « extension= amazon-elasticache-cluster-client .so » dans le fichier :

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php.d/memcached.ini
```

6. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 5 sur une AMI de Red Hat Enterprise Linux 7,0 (64 bits et 32 bits)

1. Lancez une instance Red Hat Enterprise Linux (32 bits ou 64 bits) et connectez-vous.
2. Installez les dépendances PHP :

```
sudo yum install gcc-c++ php php-pear
```

3. Téléchargez le `php-memcached` package correspondant à votre EC2 instance Amazon et à votre version de PHP. Pour de plus amples informations, veuillez consulter Téléchargement du package d'installation.
4. Installer `php-memcached`. L'URI doit être le chemin de téléchargement du package d'installation :

```
sudo pecl install <package download path>
```

5. Avec root/sudo autorisation, ajoutez un nouveau fichier nommé `memcached.ini` dans le `/etc/php.d` répertoire et insérez-le `extension=amazon-elasticache-cluster-client.so` dans le fichier.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php.d/memcached.ini
```

6. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 5 sur une AMI de serveur Ubuntu 14.04 LTS (64 bits et 32 bits)

1. Lancez une instance Ubuntu Linux (32 bits ou 64 bits) et connectez-vous.
2. Installez les dépendances PHP :

```
sudo apt-get update
sudo apt-get install gcc g++ php5 php-pear
```

3. Téléchargez le `php-memcached` package correspondant à votre EC2 instance Amazon et à votre version de PHP. Pour de plus amples informations, veuillez consulter Téléchargement du package d'installation.
4. Installer `php-memcached`. L'URI doit être le chemin de téléchargement du package d'installation.

```
sudo pecl install <package download path>
```

> **Note**
>
> Cette étape permet d'installer l'artefact de build `amazon-elasticache-cluster-client.so` dans le répertoire `/usr/lib/php5/20121212*`. Vérifiez le chemin d'accès absolu de l'artefact de génération, car vous en avez besoin pour l'étape suivante.

Si la commande précédente ne fonctionne pas, vous devez extraire manuellement l'artefact de client PHP `amazon-elasticache-cluster-client.so` du fichier `*.tgz` téléchargé, et le copier dans l'annuaire `/usr/lib/php5/20121212*`.

```
tar -xvf <package download path>
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Avec root/sudo autorisation, ajoutez un nouveau fichier nommé `memcached.ini` dans le `/etc/php5/cli/conf.d` répertoire et insérez « extension=<absolute path to amazon-elasticache-cluster-client .so> » dans le fichier.

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee
 --append /etc/php5/cli/conf.d/memcached.ini
```

6. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 5 pour AMI de SUSE Linux Enterprise Server 11 (64 bits ou 32 bits)

1. Lancez une instance SUSE Linux (32 bits ou 64 bits) et connectez-vous.
2. Installez les dépendances PHP :

```
sudo zypper install gcc php53-devel
```

3. Téléchargez le `php-memcached` package correspondant à votre EC2 instance Amazon et à votre version de PHP. Pour de plus amples informations, veuillez consulter Téléchargement du package d'installation.
4. Installer `php-memcached`. L'URI doit être le chemin de téléchargement du package d'installation.

```
sudo pecl install <package download path>
```

5. Avec root/sudo autorisation, ajoutez un nouveau fichier nommé `memcached.ini` dans le `/etc/php5/conf.d` répertoire et insérez-le **`extension=amazon-elasticache-cluster-client.so`** dans le fichier.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php5/conf.d/memcached.ini
```

6. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

> **Note**
>
> Si l'étape 5 ne fonctionne pas pour une des plateformes précédentes, vérifiez le chemin d'installation de `amazon-elasticache-cluster-client.so`. Spécifiez également le chemin d'accès complet du fichier binaire dans l'extension. En outre, vérifiez que la version du PHP en cours d'utilisation est prise en charge. Nous prenons en charge les versions 5.3 via 5.5.

Autres distributions Linux

Sur certains systèmes, notamment Cent OS7 et Red Hat Enterprise Linux (RHEL), la version 7.1 `libsasl2.so.3` a été remplacée`libsasl2.so.2`. Sur ces systèmes, lorsque vous chargez le client du ElastiCache cluster, celui-ci tente de le trouver et de le charger, mais il échoue`libsasl2.so.2`. Pour résoudre ce problème, créez un lien symbolique vers `libsasl2.so.3` afin que lorsque le client tente de charger libsasl2.so.2, il est redirigé vers `libsasl2.so.3`. Le code suivant crée ce lien symbolique.

```
cd /usr/lib64
sudo ln libsasl2.so.3 libsasl2.so.2
```

Suppression du client de cluster PHP

Rubriques

- Suppression d'une version antérieure de PHP 7
- Suppression d'une version antérieure de PHP 5

Suppression d'une version antérieure de PHP 7

Pour supprimer une version antérieure de PHP 7

1. Supprimez le fichier `amazon-elasticache-cluster-client.so` dans le répertoire de lib PHP approprié comme indiqué précédemment dans les instructions d'installation. Consultez la section pour votre installation dans Pour les utilisateurs qui ont déjà l'extension php-memcached installée.

2. Supprimez la ligne `extension=amazon-elasticache-cluster-client.so` du fichier `php.ini`.
3. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Suppression d'une version antérieure de PHP 5

Pour supprimer une version antérieure de PHP 5

1. Supprimez l'extension `php-memcached` :

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Supprimez le fichier `memcached.ini` ajouté dans le répertoire approprié comme indiqué dans les étapes d'installation précédentes.

Compiler le code source du client de ElastiCache cluster pour PHP

Cette section explique comment obtenir et compiler le code source du client de ElastiCache cluster pour PHP.

Il existe deux paquets que vous devez extraire GitHub et compiler : aws-elasticache-cluster-client-libmemcached et -. aws-elasticache-cluster-client memcached-for-php

Rubriques

- Compilation de la bibliothèque libmemcached
- Compilation du client de découverte automatique ElastiCache Memcached pour PHP

Compilation de la bibliothèque libmemcached

Pour compiler la aws-elasticache-cluster-client bibliothèque -libmemcached

1. Lancez une EC2 instance Amazon.
2. Installez les dépendances de bibliothèque.
   - Sur AMI d'Amazon Linux 201509

     ```
     sudo yum install gcc gcc-c++ autoconf libevent-devel
     ```

   - Sur AMI d'Ubuntu 14.04

     ```
     sudo apt-get update
     sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
     ```

3. Extrayez le référentiel et compilez le code.

   ```
   Download and install https://github.com/awslabs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz
   ```

Compilation du client de découverte automatique ElastiCache Memcached pour PHP

Les sections suivantes décrivent comment compiler le client ElastiCache Memcached Auto Discovery

Rubriques

- Compiler le client ElastiCache Memcached pour PHP 7

- Compiler le client ElastiCache Memcached pour PHP 5

Compiler le client ElastiCache Memcached pour PHP 7

Exécutez l'ensemble de commandes dans le répertoire du code.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-
php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-
memcached-sasl
make
make install
```

> **Note**
>
> Vous pouvez lier statiquement la bibliothèque libmemcached dans le binaire PHP afin qu'il soit porté sur différentes plateformes Linux. Pour cela, exécutez la commande suivante avant `make` :
>
> ```
> sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -
> lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
> ```

Compiler le client ElastiCache Memcached pour PHP 5

Compilez le `aws-elasticache-cluster-client-memcached-for-php` en exécutant les commandes suivantes dans le dossier `aws-elasticache-cluster-client-memcached-for-php/`.

```
git clone https://github.com/awslabs/aws-elasticache-cluster-client-memcached-for-
php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
```

```
make install
```

Configuration des ElastiCache clients

Un ElastiCache cluster est conforme au protocole avec Valkey, Memcached et Redis OSS. Le code, les applications et les outils les plus courants que vous utilisez aujourd'hui dans votre environnement existant fonctionneront parfaitement avec le service.

Cette section contient des considérations spécifiques pour la connexion aux nœuds de cache dans ElastiCache.

Rubriques

- Commandes limitées
- Recherche des points de terminaison et des numéros de port des nœuds
- Connexion pour l'utilisation de la découverte automatique
- Connexion aux nœuds d'un cluster Valkey ou Redis OSS
- Noms DNS et réseau IP sous-jacent

Commandes limitées

Pour offrir une expérience de service géré, ElastiCache restreint l'accès à certaines commandes spécifiques au moteur de cache qui nécessitent des privilèges avancés. Pour les clusters de cache exécutant Redis OSS, les commandes suivantes ne sont pas disponibles :

- `bgrewriteaof`
- `bgsave`
- `config`
- `debug`
- `migrate`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Recherche des points de terminaison et des numéros de port des nœuds

Pour se connecter à un nœud de cache, votre application a besoin de connaître le point de terminaison et le numéro de port de ce nœud.

Recherche des points de terminaison et des numéros de port des nœuds (console)

Pour déterminer les points de terminaison et les numéros de port des nœuds de

1. Connectez-vous à la console ElastiCache de gestion Amazon et choisissez le moteur exécuté sur votre cluster.

   La liste de tous les clusters exécutant le moteur choisi s'affiche.
2. Poursuivez ci-dessous pour le moteur et la configuration que vous exécutez.
3. Choisissez le nom du cluster qui vous intéresse.
4. Recherchez les colonnes Port et Endpoint pour le nœud qui vous intéresse.

Recherche des points de terminaison et des numéros de port des nœuds de cache (AWS CLI)

Pour déterminer les points de terminaison et les numéros de port des nœuds de cache, utilisez la commande `describe-cache-clusters` avec le paramètre `--show-cache-node-info`.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Les noms DNS complets et les numéros de port sont dans la section Endpoint du résultat.

Recherche des points de terminaison et des numéros de port des nœuds de cache (ElastiCache API)

Pour déterminer les points de terminaison et les numéros de port des nœuds de cache, utilisez l'action `DescribeCacheClusters` avec le paramètre `ShowCacheNodeInfo=true`.

Example

```
https://elasticache.us-west-2.amazonaws.com /
    ?Action=DescribeCacheClusters
    &ShowCacheNodeInfo=true
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
```

```
    &Timestamp=20140421T220302Z
    &Version=2014-09-30
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20140421T220302Z
    &X-Amz-Expires=20140421T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

Connexion pour l'utilisation de la découverte automatique

Si vos applications utilisent la découverte automatique, il vous suffit de connaître le point de terminaison de configuration du cluster, plutôt que les points de terminaison individuels de chaque nœud de cache. Pour de plus amples informations, veuillez consulter Identifiez automatiquement les nœuds de votre cluster (Memcached).

> **Note**
>
> Pour l'instant, la découverte automatique n'est disponible que pour les clusters de cache sous Memcached.

Connexion aux nœuds d'un cluster Valkey ou Redis OSS

> **Note**
>
> À l'heure actuelle, les clusters (API/CLI : groupes de réplication) qui prennent en charge la réplication et la lecture des répliques ne sont pris en charge que pour les clusters exécutant Valkey ou Redis OSS.

Pour les clusters, ElastiCache fournit des interfaces de console, de CLI et d'API pour obtenir des informations de connexion pour les nœuds individuels.

Pour l'activité de lecture seule, les applications peuvent se connecter à n'importe quel nœud du cluster. Toutefois, pour les activités d'écriture, nous recommandons que vos applications se connectent au point de terminaison principal (Valkey ou Redis OSS (mode cluster désactivé)) ou au point de configuration (Valkey ou Redis OSS (mode cluster activé)) du cluster au lieu de se connecter directement à un nœud. Cela permet de s'assurer que vos applications trouveront toujours le nœud

correct, même si vous décidez de reconfigurer votre cluster en faisant la promotion d'un réplica en lecture en réplica principal.

Connexion aux clusters dans un groupe de réplication (console)

Pour déterminer les points de terminaison et les numéros de port

- Consultez la rubrique, Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console).

Connexion aux clusters dans un groupe de réplication (AWS CLI)

Pour déterminer les points de terminaison et les numéros de port des nœuds de cache

Utilisez la commande `describe-replication-groups` avec le nom de votre groupe de réplication :

```
aws elasticache describe-replication-groups redis2x2
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "ReplicationGroups": [
        {
            "Status": "available",
            "Description": "2 shards, 2 nodes (1 + 1 replica)",
            "NodeGroups": [
                {
                    "Status": "available",
                    "Slots": "0-8191",
                    "NodeGroupId": "0001",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2c",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0001-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0001-002"
                        }
```

```
                    ]
                },
                {
                    "Status": "available",
                    "Slots": "8192-16383",
                    "NodeGroupId": "0002",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2b",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0002-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "redis2x2-0002-002"
                        }
                    ]
                }
            ],
            "ConfigurationEndpoint": {
                "Port": 6379,
                "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
            },
            "ClusterEnabled": true,
            "ReplicationGroupId": "redis2x2",
            "SnapshotRetentionLimit": 1,
            "AutomaticFailover": "enabled",
            "SnapshotWindow": "13:00-14:00",
            "MemberClusters": [
                "redis2x2-0001-001",
                "redis2x2-0001-002",
                "redis2x2-0002-001",
                "redis2x2-0002-002"
            ],
            "CacheNodeType": "cache.m3.medium",
            "PendingModifiedValues": {}
        }
    ]
}
```

Connexion aux clusters d'un groupe de réplication (ElastiCache API)

Pour déterminer les points de terminaison et les numéros de port des nœuds de cache

Appelez `DescribeReplicationGroups` avec le paramètre suivant :

`ReplicationGroupId` = le nom de votre groupe de réplication.

Example

```
https://elasticache.us-west-2.amazonaws.com /
    ?Action=DescribeCacheClusters
    &ReplicationGroupId=repgroup01
    &Version=2014-09-30
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140421T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20140421T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20140421T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Noms DNS et réseau IP sous-jacent

Les clients gèrent une liste de serveurs contenant les adresses et les ports des serveurs contenant les données de cache. Lors de l'utilisation ElastiCache, l' DescribeCacheClusters API (ou l'utilitaire de ligne de describe-cache-clusters commande) renvoie une entrée DNS complète et un numéro de port qui peuvent être utilisés pour la liste des serveurs.

> **Important**
>
> Il est important que les applications clientes soient configurées pour résoudre fréquemment les noms DNS des nœuds de cache lorsqu'elles tentent de se connecter au point de terminaison d'un nœud de cache.

ElastiCache garantit que le nom DNS des nœuds de cache reste le même lorsque les nœuds de cache sont restaurés en cas de défaillance.

La plupart des bibliothèques clientes prennent en charge les connexions persistantes au nœud de cache, par défaut. Nous recommandons d'utiliser des connexions persistantes aux nœuds de cache lors de l'utilisation d' ElastiCache. La mise en cache de DNS côté client peut se produire à plusieurs

endroits, notamment dans les bibliothèques clientes, dans l'environnement Common Language Runtime ou le système d'exploitation client. Vous devez passer en revue la configuration de votre application à chaque couche pour vous assurer de résoudre fréquemment les adresses IP pour vos nœuds de cache.

Hiérarchisation des données ElastiCache

ElastiCache pour les clusters Valkey ou Redis OSS qui comprennent un groupe de réplication et utilisent un type de nœud de la famille r6gd, leurs données sont hiérarchisées entre la mémoire et le stockage SSD (SSD) local. La hiérarchisation des données offre une nouvelle option de rapport prix/performances pour les charges de travail Valkey ou Redis OSS en utilisant des disques SSD (SSDs) à moindre coût dans chaque nœud du cluster, en plus du stockage des données en mémoire. Elle 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 ElastiCache clusters dotés de la hiérarchisation des données, ElastiCache surveille l'heure du dernier accès à chaque élément stocké. Lorsque la mémoire disponible (DRAM) est entièrement consommée, ElastiCache utilise l'algorithme le moins récemment utilisé (LRU) pour déplacer automatiquement les éléments rarement consultés de la mémoire vers le SSD. Lorsque les données du SSD sont ultérieurement consultées, elles sont ElastiCache automatiquement et asynchrones remises 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 vous attendre à 300 microsecondes de latence supplémentaires en moyenne pour les demandes de données stockées sur le SSD par rapport aux demandes de données en mémoire.

Grâce à la plus grande taille de nœud de hiérarchisation des données (cache.r6gd.16xlarge), vous pouvez stocker jusqu'à 1 pétaoctet dans un seul cluster de 500 nœuds (500 To en utilisant 1 réplica en lecture). La hiérarchisation des données est compatible avec toutes les commandes et structures

de données Valkey ou Redis OSS prises en charge dans. ElastiCache Vous n'avez besoin d'aucune modification côté client pour utiliser cette fonction.

Rubriques

- Bonnes pratiques
- Limites
- Tarification
- Surveillance
- Utilisation de la hiérarchisation des données
- Restauration des données de la sauvegarde dans 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. 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 :

- Vous ne pouvez utiliser la hiérarchisation des données que sur des clusters faisant partie d'un groupe de réplication.
- 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-central-1`, `eu-north-1`, `eu-west-3`, `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `ap-south-1`, `ca-central-1` et `sa-east-1`.
- Vous devez utiliser un moteur Valkey 7.2 ou version ultérieure, ou un moteur Redis OSS 6.2 ou version ultérieure.

- Vous ne pouvez pas restaurer une sauvegarde d'un cluster r6gd dans un autre cluster sauf s'il utilise également r6gd.
- Vous ne pouvez pas exporter une sauvegarde vers Amazon S3 pour les clusters de hiérarchisation des données.
- La migration en ligne n'est pas prise en charge pour les clusters exécutés sur le type de nœud r6gd.
- 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). Pour de plus amples informations, veuillez consulter Dimensionnement ElastiCache.
- Le dimensionnement automatique est pris en charge sur les clusters à l'aide de la hiérarchisation des données pour Valkey version 7.2 et versions ultérieures, et Redis OSS version 7.0.7 et versions ultérieures. Pour de plus amples informations, consultez Clusters Auto Scaling Valkey et Redis OSS.
- La hiérarchisation des données prend uniquement en charge les politiques maxmemory `volatile-lru`, `allkeys-lru`, `volatile-lfu`, `allkeys-lfu` et `noeviction`.
- La sauvegarde sans formulaire est prise en charge pour Valkey version 7.2 et versions ultérieures, et pour Redis OSS version 7.0.7 et versions ultérieures. Pour de plus amples informations, veuillez consulter Implémentation de la sauvegarde et de la synchronisation.
- Les éléments de plus de 128 MiB ne sont pas déplacés vers le SSD.
- À partir de Valley 8.1 et versions ultérieures, un élément dont la taille clé+valeur est inférieure à 40 octets ne sera pas déplacé vers le SSD.

Tarification

Les nœuds R6gd ont une capacité totale 4,8 fois plus élevée (mémoire + SSD) et peuvent vous aider à réaliser des économies de plus de 60 % quand ils s'exécutent au maximum de leur capacité par rapport aux nœuds R6g (mémoire uniquement). Pour en savoir plus, consultez PricingElastiCache (Tarification).

Surveillance

ElastiCache propose des métriques conçues spécifiquement pour surveiller les clusters de performances qui utilisent la hiérarchisation des données. Pour surveiller le ratio entre les éléments en DRAM et en SSD, vous pouvez utiliser la `CurrItems` métrique de Metrics for Valkey et Redis

OSS. Vous pouvez calculer le pourcentage comme suit : (CurrItems avec Dimension : Tier = Memory * 100)/(CurrItems sans filtre de dimension).

Si la politique d'expulsion configurée le permet, elle ElastiCache commencera à expulser des éléments lorsque le pourcentage d'éléments en mémoire sera inférieur à 5 %. Sur les nœuds configurés avec une politique de non-éviction, les opérations d'écriture recevront une erreur de mémoire insuffisante.

Il est tout de même recommandé d'envisager une mise à l'échelle horizontale pour les clusters activés en mode cluster ou une mise à l'échelle supérieure pour les clusters désactivés en mode cluster lorsque le pourcentage d'éléments en mémoire tombe en dessous de 5 %. Pour plus d'informations sur la mise à l'échelle, voirMise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé). Pour plus d'informations sur les métriques pour les clusters Valkey ou Redis OSS qui utilisent la hiérarchisation des données, consultez. Métriques pour Valkey et Redis OSS

Utilisation de la hiérarchisation des données

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

Lorsque vous créez un cluster dans le cadre d'un groupe de réplication, vous utilisez la hiérarchisation des données en sélectionnant un type de nœud dans la famille r6gd, tel que cache.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 Création d'un cluster pour Valkey ou Redis OSS.

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

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

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-enabled`, l'opération échouera.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id redis-dt-cluster \
   --replication-group-description "Redis OSS cluster with data tiering" \
```

```
   --num-node-groups 1 \
   --replicas-per-node-group 1 \
   --cache-node-type cache.r6gd.xlarge \
   --engine redis \
   --cache-subnet-group-name default \
   --automatic-failover-enabled \
   --data-tiering-enabled
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id redis-dt-cluster ^
   --replication-group-description "Redis OSS cluster with data tiering" ^
   --num-node-groups 1 ^
   --replicas-per-node-group 1 ^
   --cache-node-type cache.r6gd.xlarge ^
   --engine redis ^
   --cache-subnet-group-name default ^
   --automatic-failover-enabled ^
   --data-tiering-enabled
```

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

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "redis-dt-cluster",
        "Description": "Redis OSS cluster with data tiering",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "redis-dt-cluster"
        ],
        "AutomaticFailover": "enabled",
        "DataTiering": "enabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r6gd.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Restauration des données de la sauvegarde dans des clusters avec la hiérarchisation des données activée

Vous pouvez restaurer une sauvegarde sur un nouveau cluster avec la hiérarchisation des données activée à l'aide de la (console), (AWS CLI) ou (ElastiCache API). 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 à partir de la sauvegarde dans des clusters avec la hiérarchisation des données activée (console)

Pour restaurer une sauvegarde dans un nouveau cluster avec la hiérarchisation des données activée (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes.
3. Dans la liste des sauvegardes, cochez la case située à gauche du nom de la sauvegarde à restaurer.
4. Choisissez Restore (Restaurer).
5. Renseignez la boîte de dialogue Restore Cluster. Veillez à remplir tous les champs obligatoires, ainsi que ceux dont vous ne souhaitez pas conserver la valeur par défaut.
    1. ID du cluster : obligatoire. Nom du nouveau cluster.
    2. Mode cluster activé (mise à l'échelle externe) — Choisissez cette option pour un cluster Valkey ou Redis OSS (mode cluster activé).
    3. Type de nœud : préciser cache.r6gd.xlarge ou tout autre type de nœud de la famille r6gd.
    4. Nombre de partitions : choisissez le nombre de partitions dont vous avez besoin dans le nouveau cluster (API/CLI : groupes de nœuds).
    5. Replicas per Shard (Réplicas par partition) : choisissez le nombre de nœuds de réplica en lecture souhaité dans chaque partition.
    6. Slots and keyspaces (Emplacements et espaces de clés) : choisissez la répartition des clés entre les partitions. Si vous choisissez de spécifier les répartitions de clés, remplissez le tableau en spécifiant les plages de clés de chaque partition.
    7. Zone(s) de disponibilité : spécifiez la façon dont les zones de disponibilité du cluster doivent être sélectionnées.

8. Port : modifiez cette valeur uniquement si vous souhaitez que le nouveau cluster utilise un port différent.
9. Choisir un VPC : choisissez le VPC dans lequel vous souhaitez créer ce cluster.
10. Groupe de paramètres : choisissez un groupe de paramètres qui réserve suffisamment de mémoire pour la surcharge de Valkey ou Redis OSS pour le type de nœud que vous avez sélectionné.

6. Lorsque les paramètres vous conviennent, choisissez Create.

Pour plus d'informations sur la création des clusters , consultez Création d'un cluster pour Valkey ou Redis OSS.

Restauration des données de la sauvegarde dans des clusters avec la hiérarchisation des données activée AWS CLI

Lors de la création d'un groupe de réplication 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 cache.r6gd.xlarge, et en définissant le paramètre. `--data-tiering-enabled`

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-enabled`, l'opération échouera.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id redis-dt-cluster \
   --replication-group-description "Redis OSS cluster with data tiering" \
   --num-node-groups 1 \
   --replicas-per-node-group 1 \
   --cache-node-type cache.r6gd.xlarge \
   --engine redis \
   --cache-subnet-group-name default \
   --automatic-failover-enabled \
   --data-tiering-enabled \
   --snapshot-name my-snapshot
```

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group ^
```

```
    --replication-group-id redis-dt-cluster ^
    --replication-group-description "Redis OSS cluster with data tiering" ^
    --num-node-groups 1 ^
    --replicas-per-node-group 1 ^
    --cache-node-type cache.r6gd.xlarge ^
    --engine redis ^
    --cache-subnet-group-name default ^
    --automatic-failover-enabled ^
    --data-tiering-enabled ^
    --snapshot-name my-snapshot
```

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

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "redis-dt-cluster",
        "Description": "Redis OSS cluster with data tiering",
        "Status": "creating",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "redis-dt-cluster"
        ],
        "AutomaticFailover": "enabled",
        "DataTiering": "enabled",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "06:00-07:00",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r6gd.xlarge",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

Préparation d'un cluster dans ElastiCache

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

Vous pouvez également créer un ElastiCache cluster à l'aide de AWS CloudFormation. Pour plus d'informations, consultez AWS: : : ElastiCache : CacheCluster dans le guide de l'utilisateur de AWS Cloud Formation, qui contient des conseils sur la manière de mettre en œuvre cette approche.

Chaque fois que vous créez un cluster ou un groupe de réplication , il est utile de réaliser un travail préparatoire de sorte à ne pas avoir besoin d'effectuer une mise à niveau ou d'apporter des changements immédiatement.

Rubriques

- Déterminer les exigences ElastiCache de votre cluster
- Choix de la taille de votre nœud

Déterminer les exigences ElastiCache de votre cluster

Préparation

Connaître les réponses aux questions suivantes facilite la création de votre ElastiCache cluster :

- De quel type d'instance de nœud avez-vous besoin ?

  Pour vous guider dans la sélection du type de nœud d'instance, consultez Choix de la taille de votre nœud.

- Lancez-vous votre cluster dans un cloud privé virtuel (VPC) basé sur Amazon VPC ?

  > **Important**
  >
  > Si vous envisagez de lancer votre cluster dans un VPC, vous devez créer un groupe de sous-réseaux dans le même VPC avant de commencer à créer un cluster. Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux. ElastiCache est conçu pour être accessible depuis AWS l'intérieur via Amazon EC2. Cependant, si vous lancez votre cluster dans un VPC basé sur Amazon VPC et qu'il se trouve dans un VPC, vous pouvez y accéder depuis l'extérieur d' AWS. Pour de plus amples informations, veuillez consulter Accès aux ElastiCache ressources depuis l'extérieur AWS.

- Avez-vous besoin de personnaliser les valeurs des paramètres ?

  Si vous le faites, créez un groupe de paramètres personnalisé. Pour de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

Si vous utilisez Valkey ou Redis OSS, pensez à définir `reserved-memory` ou. `reserved-memory-percent` Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

- Devez-vous créer votre propre 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 de plus amples informations, veuillez consulter Atténuation des défaillances.

Rubriques

- ElastiCache exigences en matière de mémoire et de processeur
- Configuration du cluster Memcached
- Configuration des clusters Valkey et Redis OSS
- ElastiCache exigences de dimensionnement
- ElastiCache exigences en matière d'accès
- Exigences relatives à la région, à la zone de disponibilité et à la zone locale pour ElastiCache

ElastiCache exigences en matière de mémoire et de processeur

L'élément de base d'Amazon ElastiCache est le nœud. Les nœuds sont configurés de manière individuelle ou en groupe 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.

Le moteur Memcached est en multi-thread. Le nombre de cœurs du nœud a donc un impact sur la puissance de calcul à disposition du cluster.

Configuration du cluster Memcached

ElastiCache pour les clusters Memcached sont composés de 1 à 60 nœuds. Les données contenues dans un cluster Memcached sont partitionnées parmi les nœuds du cluster. Votre application se connecte à un cluster Memcached par le biais d'une adresse de réseau appelée point de terminaison. Chaque nœud d'un cluster Memcached dispose de son propre point de terminaison que votre application utilise afin de lire ou d'écrire sur le nœud spécifique. En plus des points de terminaison de nœud, le cluster Memcached lui-même possède un point de terminaison appelé

configuration endpoint (point de terminaison de configuration). Votre application peut utiliser ce point de terminaison pour lire ou écrire sur le cluster, laissant la détermination du nœud à lire ou à écrire à la découverte automatique.

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

Configuration des clusters Valkey et Redis OSS

ElastiCache pour Valkey et Redis OSS, les clusters sont composés de 0 à 500 partitions (également appelés groupes de nœuds). Les données d'un cluster Valkey ou Redis OSS sont partitionnées entre les partitions du cluster. Votre application se connecte à un cluster Valkey ou Redis OSS à l'aide d'une adresse réseau appelée Endpoint. Les nœuds d'une partition Valkey ou Redis OSS remplissent l'un des deux rôles suivants : un nœud read/write principal et tous les autres nœuds secondaires en lecture seule (également appelés répliques en lecture). Outre les points de terminaison du nœud, le cluster Valkey ou Redis OSS lui-même possède un point de terminaison appelé point de terminaison de configuration. Votre application peut utiliser ce point de terminaison pour lire ou écrire dans le cluster, ce qui laisse à Redis OSS le soin de déterminer le nœud à ElastiCache partir duquel lire ou écrire.

Pour de plus amples informations, veuillez consulter Gestion des clusters dans ElastiCache.

ElastiCache exigences de dimensionnement

Tous les clusters peuvent augmentés par la création d'un nouveau cluster doté d'un nouveau type de nœud plus grand. Lorsque vous augmentez la taille d'un cluster Memcached, le nouveau cluster est vide au départ. Lorsque vous augmentez la taille d'un cluster Valkey ou Redis OSS, vous pouvez l'amorcer à partir d'une sauvegarde et éviter que le nouveau cluster ne démarre à vide.

Les clusters Amazon ElastiCache for Memcached peuvent être étendus ou intégrés. Pour augmenter ou diminuer un cluster Memcached, il suffit d'ajouter ou de supprimer des nœuds du cluster. Si vous avez activé la Découverte automatique et que votre application se connecte au point de terminaison de la configuration du cluster, votre application ne nécessite aucune modification lors de l'ajout ou de la suppression de nœuds.

Pour plus d'informations, consultez Dimensionnement ElastiCache dans ce guide.

ElastiCache exigences en matière d'accès

De par leur conception, les ElastiCache clusters Amazon sont accessibles à partir d' EC2 instances Amazon. L'accès réseau à un ElastiCache cluster est limité au compte qui a créé le cluster. Par conséquent, avant de pouvoir accéder à un cluster depuis une EC2 instance Amazon, vous devez autoriser l' EC2 instance Amazon à accéder au cluster. Les étapes pour ce faire varient selon que vous avez lancé EC2 -VPC ou EC2 -Classic.

Si vous avez lancé votre cluster dans EC2 -VPC, vous devez autoriser l'accès réseau au cluster. Si vous avez lancé votre cluster dans EC2 -Classic, vous devez accorder au groupe de sécurité Amazon Elastic Compute Cloud associé à l'instance l'accès à votre groupe ElastiCache de sécurité. Pour plus d'informations, consultez Étape 3. Autoriser l'accès au cluster dans ce manuel.

Exigences relatives à la région, à la zone de disponibilité et à la zone locale pour ElastiCache

Amazon ElastiCache prend en charge toutes les AWS régions. En localisant vos ElastiCache clusters dans une AWS région proche de votre application, vous pouvez réduire la latence. Si votre cluster comporte plusieurs nœuds, la localisation de vos nœuds dans différentes zones de disponibilité ou dans des zones locales peut réduire l'impact des défaillances sur votre cluster.

Pour plus d'informations, consultez les ressources suivantes :

- Choix des régions et des zones de disponibilité pour ElastiCache
- Utilisation de zones locales avec ElastiCache
- Atténuation des défaillances

Choix de la taille de votre nœud

La taille du nœud que vous sélectionnez pour votre ElastiCache cluster a un impact sur les coûts, les performances et la tolérance aux pannes.

Taille du nœud (Valkey et Redis OSS)

Pour découvrir les avantages des processeurs Graviton, consultez Processeur AWS Graviton.

Répondre aux questions suivantes peut vous aider à déterminer le type de nœud minimal dont vous avez besoin pour votre implémentation de Valkey ou Redis OSS :

- Vous attendez-vous à des charges de travail limitées en termes de débit avec plusieurs connexions client ?

  Si tel est le cas et que vous utilisez Redis OSS version 5.0.6 ou supérieure, vous pouvez obtenir un meilleur débit et une meilleure latence grâce à notre I/O fonctionnalité améliorée, qui, le cas échéantCPUs , est utilisée pour décharger les connexions client, au nom du moteur Redis OSS. Si vous utilisez Redis OSS version 7.0.4 ou supérieure, les I/O, you will get additional acceleration with enhanced I/O multiplexing, where each dedicated network IO thread pipelines commands from multiple clients into the Redis OSS engine, taking advantage of Redis OSS' ability to efficiently process commands in batches. In ElastiCache for Redis OSS v7.1 and above, we extended the enhanced I/O threads functionality to also handle the presentation layer logic. By presentation layer, what we mean is that Enhanced I/O threads améliorés ne se contentent plus de lire les entrées du client, mais aussi de les analyser au format de commande binaire Redis OSS, qui sont ensuite transmises au thread principal pour exécution, ce qui améliore les performances. Pour plus de détails, reportez-vous au billet de blog et à la page des versions prises en charge.
- Avez-vous des charges de travail qui accèdent régulièrement à un faible pourcentage de leurs données ?

  Si tel est le cas et que vous utilisez le moteur Redis OSS version 6.2 ou ultérieure, vous pouvez tirer parti de la hiérarchisation des données en choisissant le type de nœud r6gd. Avec la hiérarchisation des données, les données les moins récemment utilisées sont stockées sur le SSD. Lorsqu'il est récupéré, il y a un faible coût de latence, qui est équilibré par des économies de coûts. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

  Pour de plus amples informations, veuillez consulter Types de nœuds pris en charge.
- Quelle est la quantité totale de mémoire dont vous avez besoin pour vos données ?

Pour obtenir une estimation générale, prenez la taille des éléments que vous souhaitez mettre en cache. Multipliez cette taille par le nombre d'éléments que vous voulez conserver dans le cache en même temps. Pour obtenir une estimation raisonnable de la taille des éléments, commencez par sérialiser vos éléments de cache, puis comptez les caractères. Divisez ensuite ce nombre sur le nombre de partitions dans votre cluster.

Pour de plus amples informations, veuillez consulter Types de nœuds pris en charge.

- Quelle version de Redis OSS utilisez-vous ?

  Les versions de Redis OSS antérieures à la version 2.8.22 vous obligent à réserver davantage de mémoire pour le basculement, le snapshot, la synchronisation et la promotion d'une réplique vers les opérations principales. En effet, vous devez disposer d'une mémoire suffisante pour toutes les écritures qui se produisent au cours du processus.

  Les versions 2.8.22 et ultérieures de Redis OSS utilisent un processus de sauvegarde sans fourche qui nécessite moins de mémoire disponible que le processus précédent.

  Pour plus d'informations, consultez les ressources suivantes :

  - Implémentation de la sauvegarde et de la synchronisation
  - S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS

- S'agit-il d'une application d'écriture intensive ?

  Les applications d'écriture intensive nécessitent une plus grande capacité de mémoire disponible, la mémoire non utilisée par les données , lors de la création des instantanés ou d'un basculement. Chaque fois que le processus `BGSAVE` est exécuté, vous devez disposer d'une mémoire suffisante qui n'est pas utilisée par les données pour accueillir toutes les écritures qui se produisent au cours du processus `BGSAVE`. Par exemple, lors de la prise d'un instantané, lors de la synchronisation d'un cluster principal avec un réplica dans un cluster et lors de l'activation de la fonctionnalité de fichier annexe uniquement (AOF). Autre exemple : lors de la promotion d'un réplica en instance principale (si le mode Multi-AZ est activé). Le pire des cas est lorsque toutes vos données sont réécrites pendant le processus. Dans ce cas, vous devez disposer d'une taille d'instance de nœud avec deux fois plus de mémoire que pour les données uniquement.

  Pour en savoir plus, consultez S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS.

- Votre implémentation sera-t-elle un cluster Valkey ou Redis OSS autonome (mode cluster désactivé), ou un cluster Valkey ou Redis OSS (mode cluster activé) avec plusieurs partitions ?

  Cluster Valkey ou Redis OSS (mode cluster désactivé)

  Si vous implémentez un cluster Valkey ou Redis OSS (mode cluster désactivé), votre type de nœud doit être capable d'accueillir toutes vos données ainsi que la surcharge nécessaire, comme décrit dans le bullet précédent.

  Par exemple, supposons que vous estimez que la taille totale de tous vos articles est de 12 Go. Dans ce cas, vous pouvez utiliser un nœud `cache.m3.xlarge` avec 13,3 Go de mémoire ou un nœud `cache.r3.large` avec 13,5 Go de mémoire. Toutefois, vous aurez sans doute besoin de davantage de mémoire pour les opérations `BGSAVE`. Si votre application est très exigeante en matière d'écriture, doublez les besoins en mémoire pour atteindre au moins 24 Go. Utilisez ainsi un `cache.m3.2xlarge` disposant de 27,9 Go de mémoire ou un `cache.r3.xlarge` disposant de 30,5 Go de mémoire.

  Valkey ou Redis OSS (mode cluster activé) avec plusieurs partitions

  Si vous implémentez un cluster Valkey ou Redis OSS (mode cluster activé) avec plusieurs partitions, le type de nœud doit être capable d'accueillir des `bytes-for-data-and-overhead / number-of-shards` octets de données.

  Par exemple, supposons que vous estimez la taille totale de tous les éléments à 12 Go et que vous ayez deux partitions. Dans ce cas, vous pouvez utiliser un nœud `cache.m3.large` disposant de 6,05 Go de mémoire (12 Go/2). Toutefois, vous aurez sans doute besoin de davantage de mémoire pour les opérations `BGSAVE`. Si votre application est très exigeante en matière d'écriture, doublez les besoins en mémoire pour atteindre au moins 12 Go par partition. Utilisez ainsi un `cache.m3.xlarge` disposant de 13,3 Go de mémoire ou un `cache.r3.large` disposant de 13,5 Go de mémoire.

- Utilisez-vous des Local Zones ?

  Les Zones Locales vous permettent de placer des ressources telles qu'un ElastiCache cluster dans plusieurs emplacements proches de vos utilisateurs. Toutefois, lorsque vous choisissez la taille de votre nœud, sachez que les tailles de nœud disponibles sont limitées aux suivantes pour le moment, quelles que soient les exigences de capacité :

  - Génération actuelle :

Types de nœuds M5 : `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Types de nœuds R5: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Types de nœuds T3 : `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Pendant que votre cluster est en cours d'exécution, vous pouvez surveiller l'utilisation de la mémoire, l'utilisation du processeur, les accès au cache et les mesures relatives aux erreurs de cache publiées sur CloudWatch. Vous remarquerez peut-être que votre cluster n'a pas le taux de succès souhaité ou que les clés sont expulsées trop souvent. Dans ces cas, vous pouvez choisir une taille de nœud différente avec des spécifications de mémoire et un CPU plus grands.

Lorsque vous surveillez l'utilisation du processeur, n'oubliez pas que Valkey et Redis OSS sont à thread unique. Ainsi, multipliez l'utilisation du CPU signalée par le nombre de cœurs du processeur pour obtenir que l'utilisation réelle. Par exemple, un processeur à quatre cœurs affichant un taux d'utilisation de 20 % est en fait le processeur Redis OSS à un cœur qui fonctionne à 80 % d'utilisation.

Taille du nœud (Memcached)

Les clusters Memcached contiennent un ou plusieurs nœuds avec les données du cluster partitionnées sur les nœuds. Pour cette raison, les besoins en mémoire du cluster et la mémoire d'un nœud sont liés, mais pas identiques. Vous pouvez atteindre la capacité de mémoire de cluster souhaitée, soit en utilisant un petit nombre de nœuds de grande taille ou plusieurs nœuds de petite taille. En outre, à mesure que vos besoins changent, vous pouvez ajouter des nœuds ou en supprimer du cluster, et donc ne payer que pour ce dont vous avez besoin.

La capacité de mémoire totale de votre cluster est calculée en multipliant le nombre de nœuds du cluster par la capacité de RAM de chaque nœud après déduction de la surcharge du système. La capacité de chaque nœud est basée sur le type de nœud.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

Le nombre de nœuds de dans le cluster est un facteur clé dans la disponibilité de votre cluster Memcached. La défaillance d'un nœud simple peut avoir un impact sur la disponibilité de votre application et sur la charge de votre base de données backend. Dans ce cas, fournit un ElastiCache

remplacement pour un nœud défaillant et celui-ci est repeuplé. Pour réduire cet impact sur la disponibilité, répartissez votre mémoire et votre capacité de calcul sur un plus grand nombre de nœuds de plus petite capacité, plutôt que d'utiliser un plus petit nombre de nœuds de grande capacité.

Dans un scénario où vous avez besoin de 35 Go de mémoire cache, vous pouvez utiliser l'une des configurations suivantes :

- 11 nœuds `cache.t2.medium` avec 3,22 Go de mémoire et 2 threads chacun = 35,42 Go et 22 threads.
- 6 nœuds `cache.m4.large` avec 6,42 Go de mémoire et 2 threads chacun = 38,52 Go et 12 threads.
- 3 nœuds `cache.r4.large` avec 12,3 Go de mémoire et 2 threads chacun = 36,90 Go et 6 threads.
- 3 nœuds `cache.m4.xlarge` avec 14,28 Go de mémoire et 4 threads chacun = 42,84 Go et 12 threads.

Comparaison des options de nœuds

Type de nœud	Mémoire en Go	Cœurs	Coût horaire*	Nœuds nécessaires	Mémoire totale (en GiB)	Nombre total de cœurs	Coût mensuel
cache.t2.medium	3,22	2	0,068 USD	11	35,42	22	538,56 $
cache.m4.large	6,42	2	0,156 USD	6	38,52	12	673,92 $
cache.m4.xlarge	14,28	4	0,311 USD	3	42,84	12	671,76 USD
cache.m5.xlarge	12,93	4	0,311 USD	3	38,81	12	671,76 USD
cache.m6g.large	6,85	2	0,147$	6	41,1	12	635$

Type de nœud	Mémoire en Go	Cœurs	Coût horaire*	Nœuds nécessaires	Mémoire totale (en GiB)	Nombre total de cœurs	Coût mensuel
cache.r4.large	12.3	2	0,228 USD	3	36,9	6	492,48 USD
cache.r5.large	13,07	2	0,216 $	3	39,22	6	466,56 $
cache.r6g.large	13,07	2	0,205$	3	42,12	6	442$
* Coût horaire par nœud à compter du 8 octobre 2020.							
Coût mensuel pour une utilisation à 100 % pendant 30 jours (720 heures).							

Ces options permettent d'avoir la même capacité de mémoire, mais une capacité de calcul et un coût différents. Pour comparer les coûts de vos options spécifiques, consultez Amazon ElastiCache Pricing.

Pour les clusters Memcached, une partie de la mémoire disponible sur chaque nœud de est utilisée pour la surcharge de connexion. Pour plus d'informations, consultez Surcharge de la connexion Memcached

L'utilisation de plusieurs nœuds nécessitera la répartition des clés sur tous ces nœuds. Chaque nœud a son propre point de terminaison. Pour faciliter la gestion des terminaux, vous pouvez utiliser ElastiCache la fonction Auto Discovery, qui permet aux programmes clients d'identifier automatiquement tous les nœuds d'un cluster. Pour de plus amples informations, veuillez consulter Identifiez automatiquement les nœuds de votre cluster (Memcached).

Dans certains cas, vous n'êtes peut-être pas certain de la capacité dont vous avez besoin. Si c'est le cas, pour les tests, nous vous recommandons de commencer par un nœud `cache.m5.large`. Surveillez ensuite l'utilisation de la mémoire, l'utilisation du processeur et le taux de réussite du cache à l'aide des ElastiCache métriques publiées sur Amazon CloudWatch. Pour plus d'informations sur CloudWatch les mesures relatives à ElastiCache, voirSurveillance de l'utilisation à l'aide de CloudWatch métriques. Pour la production et des charges de travail plus grandes, les nœuds R5 offrent les meilleures performances et valeur de coût de la RAM.

Si votre cluster n'a pas le taux de réussite souhaité, vous pouvez facilement ajouter d'autres nœuds, ce qui accroît la mémoire disponible totale de votre cluster.

Si votre cluster est limité par le CPU mais s'il bénéficie d'un taux de réussite suffisant, essayez de configurer un nouveau cluster avec un type de nœud de qui fournit une plus grande puissance de calcul.

Création d'un cluster pour Valkey ou Redis OSS

Les exemples suivants montrent comment créer un cluster Valkey ou Redis OSS à l'aide de l'API AWS Management Console AWS CLI et ElastiCache .

Création d'un système d'exploitation Valkey ou Redis (mode cluster désactivé) (console)

ElastiCache prend en charge la réplication lorsque vous utilisez le moteur Valkey ou Redis OSS. Pour surveiller la latence entre le moment où les données sont écrites dans un cluster read/write principal Valkey ou Redis OSS et le moment où elles sont propagées vers un cluster secondaire en lecture seule, ElastiCache ajoute au cluster une clé spéciale,. `ElastiCacheMasterReplicationTimestamp` Cette clé est le fuseau UTC (temps universel coordonné) actuel. Étant donné qu'un cluster Valkey ou Redis OSS peut être ajouté ultérieurement à un groupe de réplication, cette clé est incluse dans tous les clusters Valkey ou Redis OSS, même s'ils ne sont initialement pas membres d'un groupe de réplication. Pour plus d'informations sur les groupes de réplication, consultez Haute disponibilité avec les groupes de réplication.

Pour créer un cluster Valkey ou Redis OSS (mode cluster désactivé), suivez les étapes décrites dans. Création d'un cluster Valkey (mode cluster désactivé) (console)

Dès que le statut de votre cluster est disponible, vous pouvez autoriser Amazon à y EC2 accéder, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 3. Autoriser l'accès au cluster et Étape 4 : Connectez-vous au nœud du 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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Si vous utilisez Redis OSS 3.2.4 ou version ultérieure, vous pouvez créer un cluster Valkey ou Redis OSS (mode cluster activé). Les clusters Valkey ou Redis OSS (mode cluster activé) prennent en charge le partitionnement de vos données sur 1 à 500 partitions (API/CLI : groupes de nœuds), mais avec certaines limites. Pour une comparaison entre Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé), voir. Moteurs et versions pris en charge

Pour créer un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de la console ElastiCache

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région dans laquelle vous souhaitez lancer ce cluster.
3. Choisissez Get started (Mise en route) dans le panneau de navigation.
4. Choisissez Create VPC (Créer un VPC) et suivez les étapes décrites à la page Création d'un cloud privé virtuel (VPC).
5. Sur la page du tableau de ElastiCache bord, choisissez Create cluster, puis Create Valkey cluster ou Create Redis OSS cluster.
6. Sous Cluster settings (Paramètres du cluster), procédez comme suit :
   a. Choisissez Configure and create a new cluster (Configurer et créer un cluster).
   b. Pour Cluster mode (Mode cluster), choisissez Enabled (Activé).
   c. Pour Cluster info (Informations sur le cluster), saisissez une valeur pour Name (Nom).
   d. (Facultatif) Saisissez une valeur pour Description.
7. Sous Location (Emplacement) :

   AWS Cloud

   1. Pour AWS Cloud, nous vous recommandons d'accepter les paramètres par défaut pour Multi-AZ et Auto-failover (Basculement automatique). Pour plus d'informations, consultez Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ.
   2. Sous Cluster settings (Paramètres de cluster)
      a. Pour Engine version (Version du moteur), choisissez une version disponible.
      b. Pour Port, utilisez le port par défaut, 6379. Si vous avez une raison d'utiliser un autre port, saisissez le numéro de port.
      c. Pour Groupe de paramètres, choisissez un groupe de paramètres ou créez-en un nouveau. Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres OSS Valkey et Redis et Création d'un groupe ElastiCache de paramètres.

> **Note**
>
> Lorsque vous sélectionnez un groupe de paramètres pour définir les valeurs de configuration du moteur, ce groupe de paramètres est appliqué à tous les clusters du magasin de données global. Dans la page Parameter Groups (Groupes de paramètres) l'attribut Global yes/no (oui/non) indique si un groupe de paramètres fait partie d'un magasin de données global.

d. Pour Type de nœud, choisissez la flèche vers le bas
(▾).
Dans la boîte de dialogue Modifier le type de nœud choisissez une valeur pour la famille d'instances pour le type de nœud souhaité. Choisissez ensuite le type de nœud que vous souhaitez utiliser pour ce cluster, puis choisissez Enregistrer.

   Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

   Si vous choisissez un type de nœud r6gd, la hiérarchisation des données est automatiquement activée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

e. Pour Nombre de partitions, choisissez le nombre de partitions (partitions/groupes de nœuds) que vous souhaitez pour ce cluster Valkey ou Redis OSS (mode cluster activé).

   Pour certaines versions de Valkey ou Redis OSS (mode cluster activé), vous pouvez modifier dynamiquement le nombre de partitions de votre cluster :

   - Redis OSS 3.2.10 et versions ultérieures : si votre cluster exécute Redis OSS 3.2.10 ou versions ultérieures, vous pouvez modifier le nombre de partitions de votre cluster de manière dynamique. Pour de plus amples informations, veuillez consulter Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).
   - Autres versions de Redis OSS — Si votre cluster exécute une version de Redis OSS antérieure à la version 3.2.10, il existe une autre approche. Pour modifier le nombre de fragments dans votre cluster dans ce cas, créez un nouveau cluster avec le nouveau nombre de fragments. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.

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

   Les restrictions suivantes existent pour Valkey ou Redis OSS (mode cluster activé).

   - 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.
   - Le nombre de réplicas de lecture par partition est fixe et ne peut pas être modifié. Si vous avez besoin de davantage ou de moins de réplicas par partition (API/CLI : groupe de nœuds), vous devez créer un cluster avec le nouveau nombre de réplicas. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.

3. Sous Connectivity (Connectivité)

   a. Dans Network type (Type de réseau), choisissez la ou les versions d'adresses IP que ce cluster prendra en charge.

   b. Pour les groupes de sous-réseaux, choisissez le sous-réseau que vous souhaitez appliquer à ce cluster. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds. ElastiCache les clusters nécessitent un sous-réseau à double pile auquel les deux IPv6 adresses sont attribuées pour fonctionner en mode double pile IPv4 et un sous-réseau « uniquement » pour fonctionner en tant que « IPv6 -only ». IPv6

      Lorsque vous créez un nouveau groupe de sous-réseaux, entrez le VPC ID (ID du VPC) auquel il appartient.

      Sélectionnez un Discovery IP type (Type d'adresse IP de découverte). Seules les adresses IP du protocole que vous avez choisi sont renvoyées.

      Pour plus d'informations, consultez :

      - Choix d'un type de réseau dans ElastiCache.
      - Créer un sous-réseau dans votre VPC

      Si vous utilisez Utilisation de zones locales avec ElastiCache, vous devez créer ou choisir un sous-réseau qui se trouve dans la zone locale.

Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux.

4. Pour Availability zone placements (Placement de zones de disponibilité), vous avez deux options :

   - Aucune préférence : ElastiCache choisit la zone de disponibilité.
   - Specify availability zones (Spécifier les zones de disponibilité) : vous spécifiez la zone de disponibilité pour chaque cluster.

     Si vous avez choisi de spécifier les Zones de disponibilité, pour chaque cluster de chaque partition, choisissez la Zone de disponibilité depuis la liste.

   Pour de plus amples informations, veuillez consulter Choix des régions et des zones de disponibilité pour ElastiCache.

5. Choisissez Next (Suivant)
6. Sous Paramètres avancés de Valkey ou Paramètres avancés de Redis OSS ou

   - Pour Security (Sécurité) :

     i. 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 autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour de plus amples informations, veuillez consulter Utilisation clés AWS KMS gérées par le client.

     - Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour Valkey 7.2 et versions ultérieures ou Redis OSS 6.0 et versions ultérieures, si vous activez le chiffrement en transit, vous serez invité à spécifier l'une des options de contrôle d'accès suivantes :

- No Access Control (Aucun contrôle d'accès) – il s'agit du paramètre par défaut. Cela indique qu'aucune restriction n'est imposée à l'accès des utilisateurs au cluster.
- User Group Access Control List (Liste de contrôle d'accès au groupe d'utilisateurs) : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs pouvant accéder au cluster. Pour de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.
- Utilisateur par défaut AUTH : mécanisme d'authentification pour un serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.

- AUTH — Un mécanisme d'authentification pour le serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.

> **Note**
>
> Pour les versions de Redis OSS supérieures à 3.2.6, à l'exception de la version 3.2.10, AUTH est la seule option.

ii. 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.

7. Pour des sauvegardes automatiques régulières, choisissez Activer les sauvegardes automatiques, puis entrez le nombre de jours pendant lesquels vous souhaitez conserver une sauvegarde automatique avant sa suppression automatique. Si vous ne souhaitez pas de sauvegardes automatiques régulières, désactivez la case à cocher Enable automatic backups. Dans les deux cas, vous avez toujours la possibilité de créer des sauvegardes manuelles.

   Pour plus d'informations sur la sauvegarde et la restauration, consultezInstantané et restauration.

8. (Facultatif) Spécifiez une fenêtre de maintenance. La fenêtre de maintenance indique le moment, généralement d'une durée d'une heure, chaque semaine, où ElastiCache

planifie la maintenance du système pour votre cluster. Vous pouvez autoriser ElastiCache à choisir le jour et l'heure de la fenêtre de maintenance (Aucune préférence) ou vous pouvez choisir vous-même le jour, l'heure et la durée (Specify maintenance window (Spécifier 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 de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

9. (Facultatif) Pour Logs (Journaux) :
   - Sous Log format (Format de journal), sélectionnez Text (Texte) ou JSON.
   - Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
   - Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatch journaux ou le nom de votre flux Firehose, ou choisissez Select existing, puis choisissez le nom de votre groupe de journaux de CloudWatch journaux ou le nom de votre stream Firehose,
10. Pour les balises, pour vous aider à gérer vos clusters et autres ElastiCache ressources, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Pour plus d'informations, consultez Marquer vos ressources ElastiCache .
11. Choisissez Suivant.
12. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous êtes prêt, choisissez Create (Créer).

On premises

1. Pour On premises (Sur site), nous vous recommandons de laisser Auto-failover (Basculement automatique) activé. Pour plus d'informations, voir Minimiser les temps d'arrêt ElastiCache pour Redis OSS avec Multi-AZ
2. Suivez les étapes décrites dans Utilisation d'Outposts.

Pour créer l'équivalent à l'aide de l' ElastiCache API ou à la AWS CLI place de la ElastiCache console, consultez ce qui suit :

- API : CreateReplicationGroup
- CLI : create-replication-group

Dès que l'état de votre cluster est disponible, vous pouvez autoriser EC2 l'accès à celui-ci, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez Étape 3. Autoriser l'accès au cluster et Étape 4 : Connectez-vous au nœud du 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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster (AWS CLI)

Pour créer un cluster à l'aide de AWS CLI, utilisez la `create-cache-cluster` commande.

> **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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster (CLI) Valkey ou Redis OSS (mode cluster désactivé)

Example — Un cluster Valkey ou Redis OSS (mode cluster désactivé) sans répliques en lecture

Le code CLI suivant crée un cluster de cache Valkey ou Redis OSS (mode cluster désactivé) sans répliques.

> **Note**
>
> Lors de la création d'un cluster à l'aide d'un type de nœud de la famille r6gd, vous devez passer le paramètre `data-tiering-enabled`.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine redis \
--num-cache-nodes 1 \
--cache-parameter-group default.redis6.x \
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine redis ^
```

```
--num-cache-nodes 1 ^
--cache-parameter-group default.redis6.x ^
--snapshot-arns arn:aws:s3:::amzn-s3-demo-bucket/snapshot.rdb
```

Création d'un cluster Valkey ou Redis OSS (mode cluster activé) ()AWS CLI

Les clusters Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupes de réplication) ne peuvent pas être créés à l'aide de cette opération. `create-cache-cluster` Pour créer un cluster Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupe de réplication), consultez. Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro ()AWS CLI

Pour plus d'informations, consultez AWS CLI la rubrique ElastiCache de référence `create-replication-group`.

Création d'un cluster pour Valkey ou Redis OSS (API) ElastiCache

Pour créer un cluster à l'aide de l' ElastiCache API, utilisez l'`CreateCacheCluster`action.

> **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. Consultez Supprimer un cluster dans ElastiCache.

Rubriques

- Création d'un cluster de cache (API) Valkey ou Redis OSS (ElastiCache mode cluster désactivé)
- Création d'un cluster de cache dans Valkey ou Redis OSS (mode cluster activé) (ElastiCache API)

Création d'un cluster de cache (API) Valkey ou Redis OSS (ElastiCache mode cluster désactivé)

Le code suivant crée un cluster de cache (ElastiCache API) Valkey ou Redis OSS (mode cluster désactivé).

Des sauts de ligne sont ajoutés pour faciliter la lecture.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=CreateCacheCluster
    &CacheClusterId=my-cluster
```

```
    &CacheNodeType=cache.r4.large
    &CacheParameterGroup=default.redis3.2
    &Engine=redis
    &EngineVersion=3.2.4
    &NumCacheNodes=1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb
    &Timestamp=20150508T220302Z
    &Version=2015-02-02
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20150508T220302Z
    &X-Amz-Expires=20150508T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Signature=<signature>
```

Création d'un cluster de cache dans Valkey ou Redis OSS (mode cluster activé) (ElastiCache API)

Les clusters Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupes de réplication) ne peuvent pas être créés à l'aide de cette opération. `CreateCacheCluster` Pour créer un cluster Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupe de réplication), consultez. Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro (ElastiCache API)

Pour plus d'informations, consultez la rubrique de référence sur les ElastiCache API `CreateReplicationGroup`.

Création d'un cluster pour Memcached

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

Création d'un cluster Memcached (console)

Lorsque vous utilisez le moteur Memcached, Amazon ElastiCache prend en charge le partitionnement horizontal de vos données sur plusieurs nœuds. Memcached permet la découverte automatique afin que vous n'avez pas besoin de suivre les points de terminaison de chaque nœud. Memcached suit le point de terminaison de chaque nœud, en mettant à jour la liste des points de terminaison au fur et à mesure que des nœuds sont ajoutés et supprimés. Tout ce dont votre application a besoin pour interagir avec le cluster est le point de terminaison de configuration.

Pour créer un cluster Memcached via la console, suivez les étapes décrites dans Création d'un cluster Valkey (mode cluster désactivé) (console). Lorsque vous atteignez l'étape 5, sélectionnez Créer un cache Memcached.

Dès que le statut de votre cluster est disponible, vous pouvez autoriser Amazon à y EC2 accéder, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez les étapes similaires Étape 3. Autoriser l'accès au cluster etÉtape 4 : Connectez-vous au nœud du 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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster (AWS CLI)

Pour créer un cluster à l'aide de AWS CLI, utilisez la `create-cache-cluster` commande.

> **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. Consultez Supprimer un cluster dans ElastiCache.

Création d'un cluster de cache Memcached (AWS CLI)

Le code CLI suivant crée un cluster de cache Memcached avec 3 nœuds.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
--cache-cluster-id my-cluster \
--cache-node-type cache.r4.large \
--engine memcached \
--engine-version 1.4.24 \
--cache-parameter-group default.memcached1.4 \
--num-cache-nodes 3
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine memcached ^
--engine-version 1.4.24 ^
--cache-parameter-group default.memcached1.4 ^
--num-cache-nodes 3
```

Création d'un cluster pour Memcached (API) ElastiCache

Pour créer un cluster à l'aide de l' ElastiCache API, utilisez l'`CreateCacheCluster`action.

> **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. Consultez Supprimer un cluster dans ElastiCache.

Rubriques

- Création d'un cluster de cache Memcached (API) ElastiCache

Création d'un cluster de cache Memcached (API) ElastiCache

Le code suivant crée un cluster Memcached avec 3 nœuds (ElastiCache API).

Des sauts de ligne sont ajoutés pour faciliter la lecture.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=CreateCacheCluster
    &CacheClusterId=my-cluster
    &CacheNodeType=cache.r4.large
    &Engine=memcached
    &NumCacheNodes=3
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150508T220302Z
    &Version=2015-02-02
```

```
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20150508T220302Z
    &X-Amz-Expires=20150508T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Signature=<signature>
```

Afficher les détails d'un ElastiCache cluster

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

Afficher les détails d'un cluster Memcached (console)

Vous pouvez consulter les détails d'un cluster Memcached à l'aide de la ElastiCache console, du AWS CLI for ElastiCache ou de l' ElastiCache API.

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

Pour afficher les détails d'un cluster Memcached

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste en haut à droite, choisissez la AWS région qui vous intéresse.
3. Dans le tableau de bord ElastiCache du moteur, choisissez Memcached. Cela affiche une liste de tous vos clusters qui s'exécutent sur le moteur Memcached.
4. Pour voir les détails d'un cluster, choisissez la zone à gauche du nom du cluster.
5. Pour afficher les informations sur les nœuds, choisissez l'onglet Nodes (Nœuds) qui affiche des informations sur le statut et le point de terminaison du ou des nœuds.
6. Choisissez l'onglet Metrics (Métriques) qui affiche les métriques pertinentes pour tous les nœuds du cluster. Pour de plus amples informations, consultez Surveillance de l'utilisation à l'aide de CloudWatch métriques.
7. Choisissez l'onglet Network and security (Réseau et sécurité) pour afficher des détails sur la connectivité réseau et la configuration du groupe de sous-réseaux du cluster, ainsi que sur le groupe de sécurité VPC. Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux.
8. Choisissez l'onglet Maintenance pour afficher des détails sur les paramètres de maintenance du cluster. Pour de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.
9. Choisissez l'onglet Tags (Balises) pour afficher des détails sur les balises appliquées aux ressources du cluster. Pour de plus amples informations, veuillez consulter Marquer vos ressources ElastiCache .

Affichage des détails de Valkey ou Redis OSS (mode cluster désactivé) (console)

Vous pouvez afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé) à l'aide de la ElastiCache console, du AWS CLI for ElastiCache ou de l' ElastiCache API.

La procédure suivante explique comment afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé) à l'aide de la ElastiCache console.

Pour afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé)

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le tableau de bord ElastiCache du moteur, choisissez Valkey ou Redis OSS pour afficher la liste de tous vos clusters exécutés sur ce moteur.
3. Pour voir les détails d'un cluster, cochez la case à gauche du nom du cluster. Assurez-vous de sélectionner un cluster exécutant le moteur Valkey ou Redis OSS, et non Clustered Valkey ou Clustered Redis OSS. Cette opération permet d'afficher des détails sur le cluster, y compris son point de terminaison principal.
4. Pour afficher les informations de nœud :
   a. Choisissez le nom du cluster.
   b. Choisissez l'onglet Shards and nodes (Partitions et nœuds). Cette opération permet d'afficher des détails sur chaque nœud, y compris son point de terminaison que vous devez utiliser pour lire le cluster.
5. Choisissez l'onglet Metrics (Métriques) qui affiche les métriques pertinentes pour tous les nœuds du cluster. Pour de plus amples informations, consultez Surveillance de l'utilisation à l'aide de CloudWatch métriques.
6. Pour afficher les journaux, choisissez l'onglet Logs (Journaux) qui indique si le cluster utilise des journaux lents ou des journaux de moteur et fournit des informations pertinentes. Pour de plus amples informations, veuillez consulter Livraison de journaux.
7. Choisissez l'onglet Network and security (Réseau et sécurité) pour afficher des détails sur la connectivité réseau et la configuration du groupe de sous-réseaux du cluster. Pour de plus amples informations, veuillez consulter Sous-réseaux et groupes de sous-réseaux.
8. Choisissez l'onglet Maintenance pour afficher des détails sur les paramètres de maintenance du cluster. Pour de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

9. Choisissez l'onglet Service updates (Mises à jour du service) pour afficher des détails sur les mises à jour de service disponibles, ainsi que leur date limite d'application recommandée. Pour de plus amples informations, veuillez consulter Mises à jour du service dans ElastiCache.
10. Choisissez l'onglet Tags (Balises) pour afficher des détails sur les balises appliquées aux ressources du cluster. Pour de plus amples informations, veuillez consulter Marquer vos ressources ElastiCache .

Affichage des détails d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Vous pouvez afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de la ElastiCache console, du AWS CLI for ElastiCache ou de l' ElastiCache API.

La procédure suivante explique comment afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de la ElastiCache console.

Pour afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster activé)

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste en haut à droite, choisissez la AWS région qui vous intéresse.
3. Dans le tableau de bord ElastiCache du moteur, choisissez Valkey ou Redis OSS pour afficher la liste de tous vos clusters exécutés sur ce moteur.
4. Pour voir les détails d'un cluster Valkey ou Redis OSS (mode cluster activé), cochez la case à gauche du nom du cluster. Assurez-vous de choisir un cluster exécutant le moteur Valkey ou le moteur Redis OSS en cluster.

   L'écran se développe sous le cluster et affiche des détails sur le cluster, y compris le point de terminaison de configuration du cluster.
5. Pour afficher une liste des partitions du cluster et le nombre de nœuds dans chaque partition, choisissez l'onglet Shards and nodes (Partitions et nœuds).
6. Pour afficher des informations spécifiques sur un nœud :

   - Choisissez l'ID de partition.

     Cela vous permet d'afficher les informations sur chaque nœud, y compris le point de terminaison de chaque nœud que vous devez utiliser pour lire les données du cluster.

7. Choisissez l'onglet Metrics (Métriques) qui affiche les métriques pertinentes pour tous les nœuds du cluster. Pour de plus amples informations, consultez Surveillance de l'utilisation à l'aide de CloudWatch métriques.
8. Pour afficher les journaux, choisissez l'onglet Logs (Journaux) qui indique si le cluster utilise des journaux lents ou des journaux de moteur et fournit des informations pertinentes. Pour de plus amples informations, veuillez consulter Livraison de journaux.
9. Choisissez l'onglet Network and security (Réseau et sécurité) pour afficher des détails sur la connectivité réseau et la configuration du groupe de sous-réseaux du cluster, le groupe de sécurité VPC et la méthode de chiffrement activée sur le cluster, le cas échéant. Pour plus d'informations, consultez Sous-réseaux et groupes de sous-réseaux et Sécurité des données sur Amazon ElastiCache.
10. Choisissez l'onglet Maintenance pour afficher des détails sur les paramètres de maintenance du cluster. Pour de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.
11. Choisissez l'onglet Service updates (Mises à jour du service) pour afficher des détails sur les mises à jour de service disponibles, ainsi que leur date limite d'application recommandée. Pour de plus amples informations, veuillez consulter Mises à jour du service dans ElastiCache.
12. Choisissez l'onglet Tags (Balises) pour afficher des détails sur les balises appliquées aux ressources du cluster. Pour de plus amples informations, veuillez consulter Marquer vos ressources ElastiCache .

Afficher les détails d'un ElastiCache cluster (AWS CLI)

Le code suivant répertorie les informations relatives à *my-cluster* :

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

*my-cluster*Remplacez-le par le nom de votre cluster dans le cas où le cluster est créé avec 1 nœud de cache et 0 partition à l'aide de la `create-cache-cluster` commande.

```
{
    "CacheClusters": [
        {
            "CacheClusterStatus": "available",
            "SecurityGroups": [
                {
                    "Status": "active",
```

```
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "08:30-09:30",
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "CacheClusterId": "my-cluster1",
            "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
            "PreferredAvailabilityZone": "us-west-2c",
            "AuthTokenEnabled": false,
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
           "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2"
            },
            "SnapshotRetentionLimit": 0,
            "AutoMinorVersionUpgrade": true,
            "EngineVersion": "3.2.10",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        }
```

```
{
    "CacheClusters": [
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
```

```
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
            "CacheClusterId": "my-cluster2-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
```

```
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
            "CacheClusterId": "my-cluster2-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": false,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": false,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": false,
            "ReplicationGroupId": "my-cluster2",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
            "CacheClusterId": "my-cluster2-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2"
```

```
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.10",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        }
```

```
{
    "CacheClusters": [
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
            "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
```

```
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.6",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
```

```
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0001-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
```

```
            "PreferredAvailabilityZone": "us-west-2b",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-001",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2c",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-002",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
```

```
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis3.2.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "3.2.6",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
        },
        {
            "SecurityGroups": [
                {
                    "Status": "active",
                    "SecurityGroupId": "sg-dbe93fa2"
                }
            ],
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "AuthTokenEnabled": true,
            "CacheSubnetGroupName": "default",
            "SnapshotWindow": "12:30-13:30",
            "AutoMinorVersionUpgrade": true,
            "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
            "CacheClusterStatus": "available",
            "AtRestEncryptionEnabled": true,
            "PreferredAvailabilityZone": "us-west-2a",
            "TransitEncryptionEnabled": true,
            "ReplicationGroupId": "my-cluster3",
            "Engine": "redis",
            "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
            "CacheClusterId": "my-cluster3-0002-003",
            "PendingModifiedValues": {},
            "CacheNodeType": "cache.r4.large",
             "DataTiering": "disabled",
            "CacheParameterGroup": {
                "CacheNodeIdsToReboot": [],
                "ParameterApplyStatus": "in-sync",
                "CacheParameterGroupName": "default.redis6.x.cluster.on"
            },
            "SnapshotRetentionLimit": 0,
            "EngineVersion": "6.0",
            "CacheSecurityGroups": [],
            "NumCacheNodes": 1
```

```
        }
    ]
}
```

Si le cluster est créé à l'aide du AWS Management Console (nœud de cluster activé ou désactivé avec une ou plusieurs partitions), utilisez la commande suivante pour décrire les détails du cluster (remplacez *my-cluster* par le nom du groupe de réplication (nom de votre cluster) :

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée describe-cache-clusters.

Afficher les détails d'un ElastiCache cluster (ElastiCache API)

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

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

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheClusters
   &CacheClusterId=my-cluster
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Le code suivant affiche les détails de 25 clusters maximum.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheClusters
   &MaxRecords=25
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
```

```
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez la rubrique de référence sur les ElastiCache API `DescribeCacheClusters`.

Modification d'un ElastiCache cluster

Outre l'ajout ou la suppression de nœuds dans un ElastiCache cluster, il peut arriver que vous deviez apporter d'autres modifications, telles que l'ajout d'un groupe de sécurité, la modification de la fenêtre de maintenance ou d'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 changez les paramètres d'un cluster, le changement est appliqué au cluster immédiatement ou après le redémarrage du 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. Pour déterminer à quel moment une modification de paramètre particulière est appliquée, consultez la section Prise d'effet des modifications de la colonne Détails des tableaux pour Paramètres spécifiques Memcachedet Paramètres OSS Valkey et Redis. Pour plus d'informations sur le redémarrage des nœuds d'un cluster, consultez Redémarrage des nœuds.

À l'aide du ElastiCache AWS Management Console

Pour modifier un cluster

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 volet de navigation, choisissez le moteur s'exécutant sur le cluster que vous voulez modifier.

   Une liste de clusters du moteur choisi s'affiche.
4. Dans la liste des clusters, choisissez le nom du cluster que vous souhaitez modifier.
5. Choisissez Actions, puis Modifier.

   La fenêtre Modify Cluster s'affiche.
6. Dans la fenêtre Modifier un cluster, effectuez les modifications nécessaires. Les options incluent :

   - Description

- Mode cluster : pour modifier le mode cluster de Désactivé à Activé, vous devez d'abord définir le mode cluster sur Compatible.

  Le mode compatible permet à vos clients Valkey ou Redis OSS de se connecter en utilisant à la fois le mode cluster activé et le mode cluster désactivé. Après avoir migré tous les clients Valkey ou Redis OSS pour utiliser le mode cluster activé, vous pouvez terminer la configuration du mode cluster et définir le mode cluster sur Activé.
- Compatibilité de version de moteur

  > **Important**
  >
  > Vous pouvez mettre à niveau vers des versions de moteur plus récentes. Si vous mettez à niveau les versions majeures du moteur, par exemple, de 5.0.6 à 6.0, vous devez sélectionner une famille de groupe de paramètres compatible avec la nouvelle version du moteur. Pour plus d'informations sur la procédure à utiliser, consultez Gestion des versions pour ElastiCache. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant et en le recréant.
- Groupes de sécurité VPC
- Parameter Group
- 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 la rubrique Hiérarchisation des données.
- Multi-AZ
- Basculement automatique (mode cluster désactivé uniquement)
- Activer les sauvegardes automatiques
- ID du nœud de Backup
- Période de conservation de la sauvegarde
- Fenêtre de sauvegarde

- Rubrique pour la notification SNS

- Compatibilité des versions du moteur Memcached
- Network type (Type de réseau)

> **Note**
>
> Si vous passez de IPv4 à IPv6, vous devez sélectionner ou créer des groupes de sous-réseaux compatibles avec IPv6. Pour de plus amples informations, veuillez consulter Choix d'un type de réseau dans ElastiCache.

- Groupes de sécurité VPC
- Parameter Group
- Fenêtre de maintenance
- Rubrique pour la notification SNS

La case Appliquer immédiatement s'applique uniquement aux modifications de version du moteur et de type de nœud. Pour appliquer les modifications immédiatement, cochez la case Apply Immediately (Appliquer immédiatement). Si elle n'est pas cochée, les modifications apportées à la version du moteur sont appliquées lors du prochain créneau de maintenance. D'autres modifications, telles que la modification du créneau de maintenance, sont appliquées immédiatement.

Pour enable/disable enregistrer la livraison pour Redis

1. Dans la liste des clusters, choisissez le cluster que vous souhaitez augmenter. Cliquez sur l'onglet Cluster name (Nom du cluster) et pas la case à cocher à côté.
2. Sur la page des détails du cluster, choisissez l'onglet Logs.
3. Pour activer ou désactiver les journaux lents, choisissez Activer ou Désactiver.

   Si vous choisissez Enable (Activer) :

   a. Sous Log format (Format de journal), choisissez JSON ou Text (Texte).
   b. Sous Type de destination du journal, sélectionnez CloudWatch Logs ou Kinesis Firehose.

c. Sous Destination du journal, vous pouvez choisir Create new et saisir le nom de votre groupe de CloudWatchLogs journaux ou le nom de votre flux Kinesis Data Firehose. Vous pouvez également sélectionner Select existing, puis choisir le nom de votre groupe de CloudWatchLogs logs ou le nom de votre flux Kinesis Data Firehose.

d. Sélectionnez Activer.

Pour modifier votre configuration pour Redis, procédez comme suit :

1. Sélectionnez Modifier.
2. Sous Log format (Format de journal), choisissez JSON ou Text (Texte).
3. Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
4. Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatchLogs journaux ou le nom de votre flux Kinesis Data Firehose. Vous pouvez également choisir Select existing, puis le nom de votre groupe de CloudWatchLogs logs ou le nom de votre flux Kinesis Data Firehose.

En utilisant le AWS CLI avec ElastiCache

Vous pouvez modifier un cluster existant à l'aide de cette AWS CLI `modify-cache-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.

> **Important**
>
> Vous pouvez effectuer une mise à niveau vers de nouvelles versions du moteur Memcached. Pour plus d'informations sur la procédure à utiliser, consultez Gestion des versions pour ElastiCache. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant et en le recréant.

> **Important**
>
> Vous pouvez effectuer une mise à niveau vers les nouvelles versions du moteur Valkey ou Redis OSS. Si vous mettez à niveau les principales versions du moteur, par exemple de Redis OSS 5.0.6 vers Redis OSS 6.0, vous devez sélectionner une famille de groupes de

paramètres compatible avec la nouvelle version du moteur. Pour plus d'informations sur la procédure à utiliser, consultez Gestion des versions pour ElastiCache. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant et en le recréant.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Pour Windows :

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Le paramètre `--apply-immediately` s'applique uniquement à des modifications dans le type de nœud, la version du moteur et aux modifications du nombre de nœuds dans un cluster Si vous souhaitez appliquer une de ces modifications immédiatement, utilisez le paramètre `--apply-immediately`. Si vous préférez l'application de ces modifications au prochain créneau de maintenance, utilisez le paramètre `--no-apply-immediately`. D'autres modifications, telles que la modification du créneau de maintenance, sont appliquées immédiatement.

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée `modify-cache-cluster`.

Utilisation de l' ElastiCache API

Vous pouvez modifier un cluster existant à l'aide de l'`ModifyCacheCluster`opération ElastiCache API. 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.

> **Important**
>
> Vous pouvez effectuer une mise à niveau vers de nouvelles versions du moteur Memcached. Pour plus d'informations sur la procédure à utiliser, consultez Gestion des versions pour

ElastiCache. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant et en le recréant.

> **Important**
>
> Vous pouvez effectuer une mise à niveau vers les nouvelles versions du moteur Valkey ou Redis OSS. Si vous mettez à niveau les principales versions du moteur, par exemple de Redis OSS 5.0.6 vers Redis OSS 6.0, vous devez sélectionner une famille de groupes de paramètres compatible avec la nouvelle version du moteur. Pour plus d'informations sur la procédure à utiliser, consultez Gestion des versions pour ElastiCache. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant et en le recréant.

Des sauts de ligne sont ajoutés pour faciliter la lecture.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &CacheClusterId=my-cluster
    &PreferredMaintenanceWindow=sun:23:00-mon:02:00
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150901T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20150202T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20150901T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Le paramètre `ApplyImmediately` s'applique uniquement à des modifications dans le type de nœud, la version du moteur et aux modifications du nombre de nœuds dans un cluster Si vous souhaitez appliquer une de ces modifications immédiatement, définissez le paramètre `ApplyImmediately` sur `true`. Si vous préférez reporter l'application de ces modifications au prochain créneau de maintenance, définissez le paramètre `ApplyImmediately` sur `false`. D'autres modifications, telles que la modification du créneau de maintenance, sont appliquées immédiatement.

Pour plus d'informations, consultez la rubrique de référence sur les ElastiCache API `ModifyCacheCluster`.

Ajouter des nœuds à un ElastiCache cluster

L'ajout de nœuds à un cluster Memcached augmente le nombre de partitions de votre cluster. Lorsque vous modifiez le nombre de partitions dans un cluster, vous devez redéfinir le mappage de certains de vos espace de clé afin qu'ils soient mappés avec le nœud adéquat. Le remappage des espaces de clé augmente temporairement le nombre d'échecs d'accès au cache sur le cluster. Pour de plus amples informations, veuillez consulter Configuration de votre ElastiCache client pour un équilibrage de charge efficace (Memcached).

Pour reconfigurer votre cluster Valkey ou Redis OSS (mode cluster activé), voir Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

Vous pouvez utiliser la console ElastiCache de gestion AWS CLI ou l' ElastiCache API pour ajouter des nœuds à votre cluster.

En utilisant le ElastiCache AWS Management Console

Si vous souhaitez ajouter un nœud à un cluster Valkey ou Redis OSS (mode cluster désactivé) à nœud unique (un cluster sans réplication activée), il s'agit d'un processus en deux étapes : ajoutez d'abord la réplication, puis ajoutez un nœud de réplique.

Rubriques

- Pour ajouter une réplication à un cluster Valkey ou Redis OSS sans partitions
- Pour ajouter des nœuds à un ElastiCache cluster (console)

La procédure suivante ajoute la réplication à un Valkey ou Redis OSS à nœud unique pour lequel la réplication n'est pas activée. Lorsque vous ajoutez la réplication, le nœud existant devient le nœud principal dans le cluster avec réplication activée. Une fois la réplication ajoutée, vous pouvez ajouter jusqu'à 5 nœuds de réplica au cluster.

Pour ajouter une réplication à un cluster Valkey ou Redis OSS sans partitions

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.

   La liste des clusters exécutant ce moteur s'affiche.

3. Choisissez le nom d'un cluster, pas la case à gauche du nom du cluster, auquel vous souhaitez ajouter des nœuds.

   Ce qui suit est vrai pour un cluster Redis OSS dont la réplication n'est pas activée :

   - Il exécute Redis OSS, et non Redis OSS en cluster.
   - Il n'a pas de partitions.

     Si le cluster a des partitions, la réplication est déjà activée dessus et vous pouvez continuer sur Pour ajouter des nœuds à un ElastiCache cluster (console).
4. Choisissez Add replication.
5. Dans Ajouter une réplication, entrez une description pour ce cluster avec réplication activée.
6. Choisissez Ajouter.

   Dès que l'état du cluster retourne à disponible, vous pouvez passer à la prochaine procédure et ajouter des réplicas au cluster.

Pour ajouter des nœuds à un ElastiCache cluster (console)

La procédure suivante peut être utilisée pour ajouter des nœuds à un cluster.

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez le moteur s'exécutant sur le cluster auquel vous voulez ajouter des nœuds.

   La liste des clusters exécutant le moteur choisi s'affiche.
3. Dans la liste des clusters, choisissez le nom du cluster auquel vous souhaitez ajouter un nœud.

   Si votre cluster est un cluster Valkey ou Redis OSS (mode cluster activé), consultez. Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

   Si votre cluster est un cluster Valkey ou Redis OSS (mode cluster désactivé) ne contenant aucune partition, effectuez d'abord les étapes décrites à. Pour ajouter une réplication à un cluster Valkey ou Redis OSS sans partitions
4. Choisissez Add node (Ajouter un nœud).
5. Saisissez les informations demandées dans la boîte de dialogue Add Node (Ajouter un nœud).

6. Choisissez le bouton Apply Immediately - Yes (Appliquer immédiatement - Oui) pour ajouter ce nœud immédiatement, ou choisissez No (Non) pour ajouter ce nœud lors du prochain créneau de maintenance du cluster.

   Impact des nouvelles demandes d'ajout et de suppression sur les demandes en attente

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
Scénario 1	Suppression	Suppression	La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de suppression en attente. Par exemple, si les nœuds 0001, 0003 et 0007 sont en attente de suppression et qu'une nouvelle demande de suppression des nœuds 0002 et 0004 est émise, seuls les nœuds 0002 et 0004 seront supprimés. Les nœuds 0001, 0003 et 0007 ne seront pas supprimés.
Scénario 2	Suppression	Création	La nouvelle demande de création, en attente ou immédiate, remplace la demande de suppression en attente. Par exemple, si les nœuds 0001, 0003 et 0007 sont en attente de suppression et qu'une nouvelle demande de création d'un nœud est émise, un nouveau nœud sera créé et les nœuds 0001, 0003 et 0007 ne seront pas supprimés.

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
Scénario 3	Création	Suppression	La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de création en attente. Par exemple, s'il y a une demande en attente de création de deux nœuds et qu'une nouvelle demande est émise pour supprimer le nœud 0003, aucun nouveau nœud ne sera créé et le nœud 0003 sera supprimé.
Scénario 4	Création	Création	La nouvelle demande de création est ajoutée à la demande de création en attente. Par exemple, s'il y a une demande en attente pour créer deux nœuds et qu'une nouvelle demande est émise pour créer trois nœuds, les nouvelles demandes sont ajoutées à la demande en attente et cinq nœuds seront créés. **Important** Si la nouvelle demande de création est définie sur Apply Immediately - Yes (Appliquer immédiatement. Oui), toutes les demandes de création sont exécutées immédiatement. Si la nouvelle demande de création est définie sur Apply Immediately - No (Appliquer immédiatement. Non), toutes les demandes de création sont en attente.

Pour déterminer les opérations en attente, choisissez l'onglet Description et vérifiez combien de créations ou de suppressions en attente sont affichées. Vous ne pouvez pas avoir à la fois des créations et des suppressions en attente.

7. Choisissez le bouton Add.

   Au bout de quelques instants, les nouveaux nœuds apparaissent dans la liste des nœuds avec le statut creating. S'ils n'apparaissent pas, actualisez la page de votre navigateur. Lorsque l'état du nœud passe à available (disponible), le nouveau nœud peut être utilisé.

En utilisant le AWS CLI avec ElastiCache

Pour ajouter des nœuds à un cluster à l'aide de AWS CLI, utilisez l' AWS CLI opération `modify-cache-cluster` avec les paramètres suivants :

- `--cache-cluster-id` ID du cluster de cache auquel vous souhaitez ajouter des nœuds.
- `--num-cache-nodes` Le paramètre `--num-cache-nodes` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification. Pour ajouter des nœuds à ce cluster, `--num-cache-nodes` doit être supérieur au nombre actuel de nœuds dans ce cluster. Si cette valeur est inférieure au nombre actuel de nœuds, ElastiCache attendez-vous à ce que le paramètre `cache-node-ids-to-remove` et une liste de nœuds soient supprimés du cluster. Pour de plus amples informations, veuillez consulter En utilisant le AWS CLI avec ElastiCache.
- `--apply-immediately` ou `--no-apply-immediately` qui spécifie s'il convient d'ajouter ces nœuds immédiatement ou lors du prochain créneau de maintenance.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --num-cache-nodes 5 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --num-cache-nodes 5 ^
```

```
    --apply-immediately
```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
    "CacheCluster": {
        "Engine": "memcached",
        "CacheParameterGroup": {
            "CacheNodeIdsToReboot": [],
            "CacheParameterGroupName": "default.memcached1.4",
            "ParameterApplyStatus": "in-sync"
        },
        "CacheClusterId": "my-cluster",
        "PreferredAvailabilityZone": "us-west-2b",
        "ConfigurationEndpoint": {
            "Port": 11211,
            "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
        },
        "CacheSecurityGroups": [],
        "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
        "AutoMinorVersionUpgrade": true,
        "CacheClusterStatus": "modifying",
        "NumCacheNodes": 2,
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "SecurityGroups": [
            {
                "Status": "active",
                "SecurityGroupId": "sg-dbe93fa2"
            }
        ],
        "CacheSubnetGroupName": "default",
        "EngineVersion": "1.4.24",
        "PendingModifiedValues": {
            "NumCacheNodes": 5
        },
        "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
        "CacheNodeType": "cache.m3.medium",
         "DataTiering": "disabled",
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubrique `modify-cache-cluster`.

En utilisant le AWS CLI avec ElastiCache

Si vous souhaitez ajouter des nœuds à un cluster Valkey ou Redis OSS existant (mode cluster désactivé) dont la réplication n'est pas activée, vous devez d'abord créer le groupe de réplication en spécifiant le cluster existant comme principal. Pour de plus amples informations, veuillez consulter Création d'un groupe de réplication à l'aide d'un cluster de cache Valkey ou Redis OSS disponible ()AWS CLI. Une fois que le groupe de réplication est disponible, vous pouvez passer au processus suivant.

Pour ajouter des nœuds à un cluster à l'aide de AWS CLI, utilisez l' AWS CLI opération `increase-replica-count` avec les paramètres suivants :

- `--replication-group-id` ID du groupe de réplication auquel vous souhaitez ajouter des nœuds.
- `--new-replica-count` indique le nombre de nœuds que vous souhaitez avoir dans ce groupe de réplication après l'application de la modification. Pour ajouter des nœuds à ce cluster, `--new-replica-count` doit être supérieur au nombre actuel de nœuds dans ce cluster.
- `--apply-immediately` ou `--no-apply-immediately` qui spécifie s'il convient d'ajouter ces nœuds immédiatement ou lors du prochain créneau de maintenance.

Pour Linux, macOS ou Unix :

```
aws elasticache increase-replica-count \
    --replication-group-id my-replication-group \
    --new-replica-count 4 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache increase-replica-count ^
    --replication-group-id my-replication-group ^
    --new-replica-count 4 ^
    --apply-immediately
```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "node-test",
```

```
        "Description": "node-test",
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "node-test-001",
            "node-test-002",
            "node-test-003",
            "node-test-004",
            "node-test-005"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "node-test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
                        "CacheClusterId": "node-test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
```

```
                    },
                    {
                        "CacheClusterId": "node-test-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "node-test-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
         "DataTiering": "disabled",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false,
        "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubrique `increase-replica-count`.

Utilisation de l' ElastiCache API

Si vous souhaitez ajouter des nœuds à un cluster Valkey ou Redis OSS existant (mode cluster désactivé) dont la réplication n'est pas activée, vous devez d'abord créer le groupe de réplication en spécifiant le cluster existant comme principal. Pour de plus amples informations, veuillez consulter Ajouter des répliques à un cluster (API) autonome Valkey ou Redis OSS (mode cluster désactivé) ElastiCache . Une fois que le groupe de réplication est disponible, vous pouvez passer au processus suivant.

Pour ajouter des nœuds à un cluster (ElastiCache API)

- Appelez l'opération d'API `IncreaseReplicaCount` avec les paramètres suivants :

- `ReplicationGroupId` ID du cluster auquel vous souhaitez ajouter des nœuds.
- `NewReplicaCount` Le paramètre `NewReplicaCount` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification. Pour ajouter des nœuds à ce cluster, `NewReplicaCount` doit être supérieur au nombre actuel de nœuds dans ce cluster. Si cette valeur est inférieure au nombre actuel de nœuds, utilisez l'API `DecreaseReplicaCount` avec le nombre de nœuds à retirer du cluster.
- `ApplyImmediately` spécifie s'il convient d'ajouter ces nœuds immédiatement ou lors du prochain créneau de maintenance.
- `Region`Spécifie la AWS région du cluster à laquelle vous souhaitez ajouter des nœuds.

L'exemple suivant montre un appel à ajouter des nœuds à un cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=IncreaseReplicaCount
    &ApplyImmediately=true
    &NumCacheNodes=4
    &ReplicationGroupId=my-replication-group
    &Region=us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez la rubrique ElastiCache API `IncreaseReplicaCount`.

Utilisation de l' ElastiCache API

Pour ajouter des nœuds à un cluster (ElastiCache API)

- Appelez l'opération d'API `ModifyCacheCluster` avec les paramètres suivants :

- `CacheClusterId` ID du cluster auquel vous souhaitez ajouter des nœuds.
- `NumCacheNodes` Le paramètre `NumCachNodes` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification. Pour ajouter des nœuds à ce cluster, `NumCacheNodes` doit être supérieur au nombre actuel de nœuds dans ce cluster. Si cette valeur est inférieure au nombre actuel de nœuds, ElastiCache attendez-vous à ce que le paramètre `CacheNodeIdsToRemove` contenant une liste de nœuds soit supprimé du cluster (voirUtilisation de l' ElastiCache API avec Memcached).
- `ApplyImmediately` spécifie s'il convient d'ajouter ces nœuds immédiatement ou lors du prochain créneau de maintenance.
- `Region`Spécifie la AWS région du cluster à laquelle vous souhaitez ajouter des nœuds.

L'exemple suivant montre un appel à ajouter des nœuds à un cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &NumCacheNodes=5
 &CacheClusterId=my-cluster
 &Region=us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez la rubrique ElastiCache API `ModifyCacheCluster`.

Supprimer des nœuds d'un ElastiCache cluster

Vous pouvez supprimer un nœud d'un cluster Valkey, Memcached ou Redis OSS à l'aide de l'API AWS Management Console, de ou de l'API AWS CLI. ElastiCache

> **Note**
>
> Chaque fois que vous modifiez le nombre de nœuds dans un cluster Memcached, vous devez redéfinir le mappage, au minimum, d'une partie de votre espace de clé vers le nœud adéquat. Pour en savoir plus sur l'équilibrage de charge d'un cluster Memcached, consultez Configuration de votre ElastiCache client pour un équilibrage de charge efficace (Memcached).

En utilisant le ElastiCache AWS Management Console

Pour supprimer des nœuds d'un cluster (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région du cluster dont vous souhaitez supprimer des nœuds.
3. Dans le volet de navigation, choisissez le moteur s'exécutant sur le cluster duquel vous voulez supprimer un nœud.

   La liste des clusters exécutant le moteur choisi s'affiche.
4. Dans la liste de clusters, choisissez le nom du cluster dans lequel vous souhaitez supprimer un nœud.

   Une liste des nœuds du cluster s'affiche.
5. Choisissez la zone de gauche de l'ID de nœud pour le nœud que vous souhaitez supprimer. À l'aide de la console ElastiCache , vous pouvez supprimer un seul nœud à la fois. Par conséquent, la sélection de plusieurs nœuds vous empêche d'utiliser le bouton Supprimer un nœud.

   La page Supprimer le nœud s'affiche.
6. Pour supprimer le nœud, remplissez la page Supprimer un nœud, puis choisissez Supprimer un nœud. Pour conserver le nœud, choisissez Annuler.

> **Important**
>
> Avec Valkey ou Redis OSS, si vous supprimez le nœud, le cluster n'est plus compatible avec Multi-AZ, assurez-vous d'abord de décocher la case Multi-AZ, puis de supprimer le nœud. Si vous désactivez la case à cocher Multi-AZ vous pouvez choisir d'activer le basculement automatique.

Impact des nouvelles demandes d'ajout et de suppression sur les demandes en attente

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
Scénario 1	Suppression	Suppression	La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de suppression en attente. Par exemple, si les nœuds 0001, 0003 et 0007 sont en attente de suppression et qu'une nouvelle demande de suppression des nœuds 0002 et 0004 est émise, seuls les nœuds 0002 et 0004 seront supprimés. Les nœuds 0001, 0003 et 0007 ne seront pas supprimés.
Scénario 2	Suppression	Création	La nouvelle demande de création, en attente ou immédiate, remplace la demande de suppression en attente. Par exemple, si les nœuds 0001, 0003 et 0007 sont en attente de suppression et qu'une nouvelle demande de création d'un nœud est émise, un nouveau nœud sera créé et les nœuds 0001, 0003 et 0007 ne seront pas supprimés.
Scénario 3	Création	Suppression	La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de création en attente.

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
			Par exemple, s'il y a une demande en attente de création de deux nœuds et qu'une nouvelle demande est émise pour supprimer le nœud 0003, aucun nouveau nœud ne sera créé et le nœud 0003 sera supprimé.
Scénario 4	Création	Création	La nouvelle demande de création est ajoutée à la demande de création en attente. Par exemple, s'il y a une demande en attente pour créer deux nœuds et qu'une nouvelle demande est émise pour créer trois nœuds, les nouvelles demandes sont ajoutées à la demande en attente et cinq nœuds seront créés. **Important** Si la nouvelle demande de création est définie sur Apply Immediately - Yes (Appliquer immédiatement. Oui), toutes les demandes de création sont exécutées immédiatement. Si la nouvelle demande de création est définie sur Apply Immediately - No (Appliquer immédiate ment. Non), toutes les demandes de création sont en attente.

Pour déterminer les opérations en attente, choisissez l'onglet Description et vérifiez combien de créations ou de suppressions en attente sont affichées. Vous ne pouvez pas avoir à la fois des créations et des suppressions en attente.

En utilisant le AWS CLI avec ElastiCache

1. Identifiez IDs les nœuds que vous souhaitez supprimer. Pour de plus amples informations, veuillez consulter Afficher les détails d'un ElastiCache cluster.

2. Utilisez l'opération `decrease-replica-count` 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 `decrease-replica-count` avec les paramètres suivants :

   - `--replication-group-id` ID du groupe de réplication auquel vous souhaitez ajouter des nœuds.
   - `--new-replica-count` Le paramètre `--new-replica-count` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification.
   - `--replicas-to-remove`Liste des nœuds IDs que vous souhaitez supprimer de ce cluster.
   - `--apply-immediately` ou `--no-apply-immediately` qui spécifie s'il convient de supprimer ces nœuds immédiatement ou lors du prochain créneau de maintenance.
   - `--region`Spécifie la AWS région du cluster dont vous souhaitez supprimer des nœuds.

   > **Note**
   >
   > Vous ne pouvez passer qu'un seul des paramètres `--replicas-to-remove` ou `--new-replica-count` lors de l'appel de cette opération.

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache decrease-replica-count \
       --replication-group-id my-replication-group \
       --new-replica-count 2 \
       --region us-east-2 \
       --apply-immediately
   ```

   Pour Windows :

   ```
   aws elasticache decrease-replica-count ^
       --replication-group-id my-replication-group ^
       --new-replica-count 3 ^
       --region us-east-2 ^
       --apply-immediately
   ```

Cette opération produit une sortie similaire à la sortie suivante (format JSON) :

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "node-test",
        "Description": "node-test"
       },
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "node-test-001",
            "node-test-002",
            "node-test-003",
            "node-test-004",
            "node-test-005",
            "node-test-006"
        ],
        "NodeGroups": [
            {
                "NodeGroupId": "0001",
                "Status": "modifying",
                "PrimaryEndpoint": {
                    "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "ReaderEndpoint": {
                    "Address": "node-test-
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                    "Port": 6379
                },
                "NodeGroupMembers": [
                    {
                        "CacheClusterId": "node-test-001",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CurrentRole": "primary"
                    },
                    {
```

```
                        "CacheClusterId": "node-test-002",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "node-test-003",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "node-test-004",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CurrentRole": "replica"
                    },
                    {
                        "CacheClusterId": "node-test-005",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    },
                    {
```

```
                        "CacheClusterId": "node-test-006",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
                            "Port": 6379
                        },
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CurrentRole": "replica"
                    }
                ]
            }
        ],
        "SnapshottingClusterId": "node-test-002",
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotRetentionLimit": 1,
        "SnapshotWindow": "07:30-08:30",
        "ClusterEnabled": false,
        "CacheNodeType": "cache.r5.large",
         "DataTiering": "disabled",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false,
        "ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
    }
}
```

Vous pouvez également appeler `decrease-replica-count` et au lieu de passer au paramètre `--new-replica-count`, vous pouvez passer le paramètre `--replicas-to-remove`, comme illustré ci-après :

Pour Linux, macOS ou Unix :

```
aws elasticache decrease-replica-count \
    --replication-group-id my-replication-group \
    --replicas-to-remove node-test-003 \
    --region us-east-2 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache decrease-replica-count ^
    --replication-group-id my-replication-group ^
    --replicas-to-remove node-test-003 ^
    --region us-east-2 ^
    --apply-immediately
```

Pour plus d'informations, consultez les AWS CLI rubriques `decrease-replica-count`.

Utilisation de l' ElastiCache API avec Valkey ou Redis OSS

Pour supprimer des nœuds à l'aide de l' ElastiCache API, appelez l'opération d'`DecreaseReplicaCount`API avec l'ID du groupe de réplication et une liste des nœuds à supprimer, comme indiqué :

- `ReplicationGroupId` ID du groupe de réplication auquel vous souhaitez ajouter des nœuds.
- `ReplicasToRemove` Le paramètre `ReplicasToRemove` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification.
- `ApplyImmediately` spécifie s'il convient de supprimer ces nœuds immédiatement ou lors du prochain créneau de maintenance.
- `Region`Spécifie la AWS région du cluster dont vous souhaitez supprimer un nœud.

L'exemple suivant supprime immédiatement les nœuds 0004 et 0005 du cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DecreaseReplicaCount
    &ReplicationGroupId=my-replication-group
    &ApplyImmediately=true
    &ReplicasToRemove=node-test-003
    &Region us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez la rubrique ElastiCache API `DecreaseReplicaCount`.

Utilisation de l' ElastiCache API avec Memcached

Pour supprimer des nœuds à l'aide de l' ElastiCache API, appelez l'opération d'`ModifyCacheClusterAPI` avec l'ID du cluster de cache et une liste des nœuds à supprimer, comme indiqué :

- `CacheClusterId` ID du cluster de cache duquel vous voulez supprimer des nœuds.
- `NumCacheNodes` Le paramètre `NumCacheNodes` spécifie le nombre de nœuds que vous voulez ajouter à ce cluster après application de la modification.
- `CacheNodeIdsToRemove.member.n`Liste des nœuds IDs à supprimer du cluster.
  - `CacheNodeIdsToRemove.member.1=0004`
  - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` spécifie s'il convient de supprimer ces nœuds immédiatement ou lors du prochain créneau de maintenance.
- `Region`Spécifie la AWS région du cluster dont vous souhaitez supprimer un nœud.

L'exemple suivant supprime immédiatement les nœuds 0004 et 0005 du cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &CacheClusterId=my-cluster
    &ApplyImmediately=true
    &CacheNodeIdsToRemove.member.1=0004
    &CacheNodeIdsToRemove.member.2=0005
    &NumCacheNodes=3
    &Region us-east-2
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez la rubrique ElastiCache API `ModifyCacheCluster`.

Annulation des opérations d'ajout ou de suppression de nœuds en attente dans ElastiCache

Si vous avez choisi de ne pas appliquer immédiatement une modification au ElastiCache cluster, l'opération est en attente jusqu'à ce qu'elle soit effectuée lors de votre prochaine fenêtre de maintenance. Vous pouvez annuler toute opération en attente.

Pour annuler une opération d'ajout ou de suppression de nœud en attente avec le AWS CLI, utilisez la `modify-cache-cluster` commande. Définissez un paramètre `num-cache-nodes` égal au nombre actuel de nœuds de cache dans le cluster, puis ajoutez l'`--apply-immediately`indicateur. Cela annulera la modification en attente.

Pour annuler l'ajout ou la suppression d'un nœud en attente :

```
aws elasticache modify-cache-cluster
 --cache-cluster-id <your-cluster-id>
 --num-cache-nodes <current-number-of-nodes>
 --apply-immediately
```

S'il n'est pas clair si des ajouts ou des suppressions de nœuds sont en attente, vous pouvez confirmer leur statut à l'aide de la `describe-cache-clusters` commande suivante :

```
aws elasticache describe-cache-clusters
 --cache-cluster-id <your-cluster-id>
```

Tous les nœuds en attente doivent apparaître dans la `PendingModifiedValues` sortie. Par exemple :

```
"PendingModifiedValues": {
 "NumCacheNodes": 3
 },
```

Supprimer un cluster dans ElastiCache

Tant qu'un ElastiCache cluster est disponible, il vous est facturé, que vous l'utilisiez activement ou non. Pour ne plus être facturé, supprimez le cluster.

> **Warning**
>
> Lorsque vous supprimez un ElastiCache cluster, vos instantanés manuels sont conservés. Vous pouvez également créer un instantané final avant la suppression du cluster. Les instantanés mis en cache automatiquement ne sont pas conservés.

En utilisant le 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 ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le ElastiCache tableau de bord du moteur, choisissez le moteur qui s'exécute dans le cluster que vous souhaitez supprimer.

   La liste de tous les clusters exécutant le moteur choisi s'affiche.
3. Pour choisir le cluster à supprimer, choisissez le nom du cluster dans la liste des clusters.

   > **Important**
   >
   > Vous ne pouvez supprimer qu'un seul cluster à la fois depuis la ElastiCache console. Le choix de plusieurs clusters désactive l'opération de suppression.
4. Pour Actions, choisissez Supprimer.
5. Dans l'écran de confirmation Supprimer le cluster, choisissez Supprimer pour supprimer le cluster ou Annuler pour le conserver.

   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.

Utilisation du AWS CLI pour supprimer un ElastiCache cluster

Le code suivant supprime le cluster `my-cluster` de ElastiCache cache.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

L'action `delete-cache-cluster` de la CLI supprime uniquement un cluster de cache. Pour supprimer plusieurs clusters de cache, appelez `delete-cache-cluster` pour chaque cluster de cache devant être supprimé. Vous n'avez pas besoin d'attendre que la suppression d'un cluster de cache soit terminée avant d'en supprimer un autre.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id my-cluster \
    --region us-east-2
```

Pour Windows :

```
aws elasticache delete-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --region us-east-2
```

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée `delete-cache-cluster`.

Utilisation de l' ElastiCache API

Le code suivant supprime le cluster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DeleteCacheCluster
    &CacheClusterId=my-cluster
    &Region us-east-2
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20150202T220302Z
```

```
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20150202T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

L'opération `DeleteCacheCluster` de la CLI supprime uniquement un cluster de cache. Pour supprimer plusieurs clusters de cache, appelez `DeleteCacheCluster` pour chaque cluster de cache devant être supprimé. Vous n'avez pas besoin d'attendre que la suppression d'un cluster de cache soit terminée avant d'en supprimer un autre.

Pour plus d'informations, consultez la rubrique de référence sur les ElastiCache API `DeleteCacheCluster`.

Accès à votre ElastiCache cluster ou à votre groupe de réplication

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

Si vous avez lancé votre ElastiCache instance dans un Amazon Virtual Private Cloud (Amazon VPC), vous pouvez accéder à votre ElastiCache instance depuis une EC2 instance Amazon située dans le même Amazon VPC. Ou bien, en utilisant le peering VPC, vous pouvez accéder à votre ElastiCache instance depuis un Amazon EC2 dans un autre Amazon VPC.

Si vous avez lancé votre ElastiCache instance dans EC2 Classic, vous autorisez l' EC2 instance à accéder à votre cluster en accordant au groupe de EC2 sécurité Amazon associé à l'instance l'accès à votre groupe de sécurité du cache. Par défaut, l'accès au cluster est limité au compte qui a lancé le cluster.

Rubriques

- Octroi de l'accès à votre cluster ou groupe de réplication

Octroi de l'accès à votre cluster ou groupe de réplication

Vous avez lancé votre cluster dans EC2 -VPC

Si vous avez lancé votre cluster dans un Amazon Virtual Private Cloud (Amazon VPC), vous ne pouvez vous ElastiCache y connecter qu'à partir d'une EC2 instance Amazon exécutée dans le même Amazon VPC. Dans ce cas, vous devez accorder l'accès au réseau au cluster.

> **Note**
>
> Si vous utilisez les zones locales (Local Zones), assurez-vous que vous les avez activées. Pour de plus amples informations, veuillez consulter Activation des zones locales. Ce faisant, votre VPC est étendu à cette zone locale et votre VPC traitera le sous-réseau comme n'importe quel sous-réseau dans toute autre zone de disponibilité et les passerelles, tables de routage et autres considérations de groupe de sécurité. sera automatiquement ajusté.

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

1. Connectez-vous à la EC2 console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.

2. Dans le volet de navigation, sous Réseau et sécurité, choisissez Groupes de sécurité.
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 à ElastiCache utiliser, 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 Memcached est **11211** Le port par défaut pour Valkey et Redis OSS est. **6379**
   e. Dans le champ Source, choisissez Anywhere doté d'une plage de ports (0.0.0.0/0) afin que toute EC2 instance Amazon que vous lancez au sein de votre Amazon VPC puisse se connecter à vos nœuds. ElastiCache

      > **Important**
      >
      > L'ouverture du ElastiCache cluster 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. Cependant, le groupe de sécurité par défaut peut être appliqué à d'autres EC2 instances Amazon du compte du client, et ces instances peuvent avoir une adresse IP publique. Si ces instances exécutent un service sur le port par défaut, ce service peut être exposé accidentellement. Nous vous recommandons donc de créer un groupe de sécurité VPC qui sera utilisé exclusivement par ElastiCache. Pour plus d'informations, consultez Groupes de sécurité personnalisés.

   f. Choisissez Enregistrer.

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

Accès aux ElastiCache ressources depuis l'extérieur AWS

Amazon ElastiCache est un AWS service qui fournit un stockage de valeurs clés en mémoire basé sur le cloud. Le service est conçu pour être accessible exclusivement de l'intérieur AWS. Toutefois, si le ElastiCache cluster est hébergé dans un VPC, vous pouvez utiliser une instance de traduction d'adresses réseau (NAT) pour fournir un accès extérieur.

Prérequis

Les conditions suivantes doivent être remplies pour que vous puissiez accéder à vos ElastiCache ressources depuis l'extérieur AWS :

- Le cluster doit se trouver dans un VPC et être accessible via une instance de traduction d'adresses réseau (NAT). Cette exigence ne souffre aucune exception.
- L'instance NAT doit être lancée dans le même VPC que le cluster.
- L'instance NAT doit être lancée dans un sous-réseau public distinct du cluster.
- Une adresse IP Elastic (EIP) doit être associée à l'instance NAT. La fonction de réacheminement d'iptables permet de rediriger un port de l'instance NAT vers le port du nœud de cache dans le cadre du VPC.

Considérations

N'oubliez pas de prendre en compte les éléments suivants lorsque vous accédez à vos ressources ElastiCache en dehors d' ElastiCache.

- Les clients se connectent au port EIP et au port cache de l'instance NAT. Le réacheminement du port de l'instance NAT redirige le trafic vers le nœud du cluster de cache approprié.
- Si un nœud de cluster est ajouté ou remplacé, les règles iptables doivent être mises à jour pour prendre en compte ce changement.

Limites

Cette approche doit être utilisée à des fins de test et de développement uniquement. Elle est déconseillée dans le cadre d'une utilisation en production en raison des limitations suivantes :

- L'instance NAT agit comme un proxy entre les clients et plusieurs clusters. L'ajout d'un proxy a une incidence négative sur les performances du cluster de cache. Ce mauvais impact augmente avec le nombre de clusters de cache auquel vous accédez via l'instance NAT.

- Le trafic depuis les clients vers l'instance NAT n'est pas chiffré. Par conséquent, vous devez éviter d'envoyer des données sensibles via l'instance NAT.
- L'instance NAT ajoute la surcharge de traitement liée à la maintenance d'une autre instance.
- L'instance NAT constitue un point de défaillance unique. Pour plus d'informations sur la façon de configurer une solution haute disponibilité des instances NAT sur VPC, veuillez consulter Solution haute disponibilité des instances NAT sur Amazon VPC : un exemple.

Comment accéder aux ElastiCache ressources depuis l'extérieur AWS

La procédure suivante explique comment vous connecter à vos ElastiCache ressources à l'aide d'une instance NAT.

Ces étapes supposent les faits suivants :

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

Vous avez ensuite besoin de l'instance NAT dans le sens opposé :

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Vous devez également activer le transfert IP, qui est désactivé par défaut :

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/
sysctl.conf sudo sysctl --system
```

- Vous accédez à un cluster Memcached avec les éléments suivants :
  - Adresse IP : 10.0.1.230
  - Port Memcached par défaut : 11211
  - Groupe de sécurité : *10\.0\.0\.55*
- Vous accédez à un cluster Valkey ou Redis OSS avec :
  - Adresse IP : 10.0.1.230
  - Port par défaut : 6379
  - Groupe de sécurité : sg-bd56b7da

- AWS adresse IP de l'instance — 198.99.100.27
- Votre client de confiance a l'adresse IP 198.51.100.27.
- Votre instance NAT a l'adresse IP Elastic 203.0.113.73.
- Votre instance NAT a le groupe de sécurité sg-ce56b7a9.

Pour vous connecter à vos ElastiCache ressources à l'aide d'une instance NAT

1. Créez une instance NAT dans le même VPC que votre cluster de cache, mais dans un sous-réseau public.

   Par défaut, l'assistant VPC lancera un type de nœud cache.m1.small. Vous devez sélectionner une taille de nœud selon vos besoins. Vous devez utiliser l'AMI EC2 NAT pour pouvoir y accéder ElastiCache depuis l'extérieur AWS.

   Pour plus d'informations sur la création d'une instance NAT, consultez la section Instances NAT dans le guide de l'utilisateur AWS VPC.
2. Créez des règles de groupe de sécurité pour le cluster de cache et l'instance NAT.

   Le groupe de sécurité de l'instance NAT et l'instance de cluster doivent avoir les règles suivantes :

   - Deux règles de trafic entrant
     - Avec Memcached, la première règle est d'autoriser les connexions TCP des clients de confiance à chaque port de cache transféré depuis l'instance NAT (11211 - 11213).
     - Avec Valkey et Redis OSS, la première règle est d'autoriser les connexions TCP des clients de confiance à chaque port de cache transféré depuis l'instance NAT (6379 - 6381).
     - Une deuxième règle pour autoriser l'accès SSH aux clients de confiance.

   Groupe de sécurité d'instances NAT - règles entrantes avec Memcached

Type	Protocole	Plage de ports	Source
Destination	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

Groupe de sécurité d'instances NAT : règles entrantes avec Valkey ou Redis OSS

Type	Protocole	Plage de ports	Source
Destination	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Avec Memcached, une règle sortante autorisant les connexions TCP au port de cache (11211).

  Groupe de sécurité de l'instance NAT - Règle de trafic sortant

Type	Protocole	Plage de ports	Destination
Destination	TCP	11211	sg-ce56b7a9 (groupe de sécurité NAT)

- Avec Valkey ou Redis OSS, une règle sortante permettant aux connexions TCP d'accéder au port de cache (6379).

  Groupe de sécurité de l'instance NAT - Règle de trafic sortant

Type	Protocole	Plage de ports	Destination
Destination	TCP	6379	sg-ce56b7a9 (groupe de sécurité NAT)

- Avec Memcached, règle entrante pour le groupe de sécurité du cluster qui autorise les connexions TCP entre l'instance NAT et le port de cache (11211).

  Groupe de sécurité de l'instance de cluster - Règle de trafic entrant

Type	Protocole	Plage de ports	Source
Destination	TCP	11211	sg-ce56b7a9 (groupe de sécurité NAT)

- Avec Valkey ou Redis OSS, règle entrante pour le groupe de sécurité du cluster qui autorise les connexions TCP entre l'instance NAT et le port de cache (6379).

  Groupe de sécurité de l'instance de cluster - Règle de trafic entrant

Type	Protocole	Plage de ports	Source
Destination	TCP	6379	sg-ce56b7a9 (groupe de sécurité NAT)

3. Validez les règles.

   - Vérifiez que le client de confiance peut accéder à l'instance NAT via SSH.
   - Vérifiez que le client de confiance peut se connecter au cluster à partir de l'instance NAT.

4. Memcached

   Ajoutez une règle iptables à l'instance NAT.

   Une règle iptables doit être ajoutée à la table NAT pour chaque nœud dans le cluster pour rediriger le port de cache depuis l'instance NAT vers le nœud de cluster. Voici un exemple possible :

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
 10.0.1.230:11211
```

   Le numéro de port doit être unique pour chaque nœud du cluster. Par exemple, si vous utilisez un cluster de Memcached de trois nœuds avec les ports 11211 – 11213, les règles devraient se présenter comme suit :

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
 10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to
 10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to
 10.0.1.232:11211
```

   Vérifiez que le client de confiance peut se connecter au cluster.

Le client de confiance doit se connecter à l'EIP associée à l'instance NAT et au port de cluster correspondant au nœud de cluster approprié. Par exemple, la chaîne de connexion pour PHP pourrait ressembler à ce qui suit :

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

Un client Telnet peut également être utilisé pour vérifier la connexion. Par exemple :

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey ou Redis OSS

Ajoutez une règle iptables à l'instance NAT.

Une règle iptables doit être ajoutée à la table NAT pour chaque nœud dans le cluster pour rediriger le port de cache depuis l'instance NAT vers le nœud de cluster. Voici un exemple possible :

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
 10.0.1.230:6379
```

Le numéro de port doit être unique pour chaque nœud du cluster. Par exemple, si vous travaillez avec un cluster Redis OSS à trois nœuds utilisant les ports 6379 à 6381, les règles seront les suivantes :

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
 10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to
 10.0.1.231:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to
 10.0.1.232:6379
```

Vérifiez que le client de confiance peut se connecter au cluster.

Le client de confiance doit se connecter à l'EIP associée à l'instance NAT et au port de cluster correspondant au nœud de cluster approprié. Par exemple, la chaîne de connexion pour PHP pourrait ressembler à ce qui suit :

```
redis->connect( '203.0.113.73', 6379 );
redis->connect( '203.0.113.73', 6380 );
redis->connect( '203.0.113.73', 6381 );
```

Un client Telnet peut également être utilisé pour vérifier la connexion. Par exemple :

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

5. Enregistrez la configuration iptables.

   Enregistrez les règles après les avoir testées et vérifiées. Si vous utilisez une distribution Linux basée sur Redhat (comme Amazon Linux), exécutez la commande suivante :

```
service iptables save
```

Rubriques en relation

Les rubriques suivantes sont susceptibles de vous intéresser.

- Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC
- Accès à un ElastiCache cache depuis une application exécutée dans le centre de données d'un client
- Instances NAT
- Configuration des ElastiCache clients
- Solution haute disponibilité des instances NAT sur Amazon VPC : un exemple

Recherche de points de terminaison de connexion dans ElastiCache

Votre application se connecte à votre ElastiCache cluster à l'aide de points de terminaison. Un point de terminaison est l'adresse unique d'un nœud ou d'un cluster.

Vous pouvez également établir une connexion privée entre votre VPC et les points de terminaison d' ElastiCache API en créant un point de terminaison VPC d'interface via. AWS PrivateLink Pour de plus amples informations, veuillez consulter ElastiCache Points de terminaison VPC de l'API et de l'interface ()AWS PrivateLink.

Quels points de terminaison utiliser avec Valkey ou Redis OSS.

- Pour un nœud autonome, utilisez le point de terminaison du nœud pour les opérations de lecture et d'écriture.

- Pour les clusters Valkey, Valkey ou Redis OSS (mode cluster désactivé), utilisez le point de terminaison principal pour toutes les opérations d'écriture. Utilisez le Point de terminaison du lecteur pour répartir également les connexions entrantes vers le point de terminaison entre toutes les réplicas lues. Utilisez les points de terminaison de nœud individuels pour les opérations d'écriture (dans l'API/la CLI, ils sont appelés points de terminaison de lecture).

- Pour les clusters Valkey ou Redis OSS (mode cluster activé), utilisez le point de terminaison de configuration du cluster pour toutes les opérations prenant en charge les commandes activées en mode cluster. Vous devez utiliser un client qui prend en charge Valkey Cluster ou Redis OSS Cluster sur Redis OSS 3.2 et versions ultérieures. Vous pouvez toujours lire à partir des points de terminaison de nœuds individuels (ils API/CLI sont appelés points de terminaison de lecture).

Les sections suivantes vous guident à travers la découverte de points de terminaison dont vous aurez besoin pour le moteur utilisé.

Quels points de terminaison utiliser avec Memcached.

Pour le cache ElastiCache sans serveur pour Memcached, il suffit d'acquérir le DNS et le port du point de terminaison du cluster depuis la console.

À partir du AWS CLI, utilisez la `describe-serverless-caches` commande pour obtenir les informations du point de terminaison.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Le résultat de l'opération ci-dessus doit ressembler à cet exemple (format JSON) :

```
{
    "ServerlessCaches": [
        {
            "ServerlessCacheName": "serverless-memcached",
            "Description": "test",
            "CreateTime": 1697659642.136,
            "Status": "available",
            "Engine": "memcached",
            "MajorEngineVersion": "1.6",
            "FullEngineVersion": "21",
            "SecurityGroupIds": [
                "sg-083eda453e1e51310"
            ],
            "Endpoint": {
                "Address": "serverless-memcached-01.amazonaws.com",
                "Port":11211
            },
            "ARN": "<the ARN>",
            "SubnetIds": [
                "subnet-0cf759df15bd4dc65",
                "subnet-09e1307e8f1560d17"
            ],
            "SnapshotRetentionLimit": 0,
            "DailySnapshotTime": "03:00"
        }
    ]
}
```

Pour un cluster Memcached basé sur une instance, si vous utilisez Automatic Discovery, vous pouvez utiliser le point de terminaison de configuration du cluster pour configurer votre client Memcached. Cela signifie que vous devez utiliser un client qui prend en charge la découverte automatique.

Si vous n'utilisez pas la découverte automatique, vous devez configurer votre client pour utiliser les points de terminaison des nœuds individuels pour les lectures et écritures. Vous devez également garder des traces de ces dernières lorsque vous ajoutez ou supprimez des nœuds.

Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)

Si un cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé) ne possède qu'un seul nœud, le point de terminaison du nœud est utilisé à la fois pour les lectures et les écritures. Si un cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé) possède plusieurs nœuds, il existe trois types de points de terminaison : le point de terminaison principal, le point de terminaison du lecteur et les points de terminaison du nœud.

Le point de terminaison principal est un nom DNS qui est toujours résolu dans le nœud principal du cluster. Le point de terminaison principal n'est pas affecté par les changements apportés à votre cluster, tels que la promotion d'un réplica en lecture au rang de réplica principal. Pour l'activité d'écriture, il est préférable que vos applications se connectent au point de terminaison principal.

Un point de terminaison de lecteur répartira de manière égale les connexions entrantes vers le point de terminaison entre toutes les répliques de lecture ElastiCache d'un cluster Redis OSS. Des facteurs supplémentaires, comme le moment où l'application crée les connexions ou la façon dont l'application utilise ou réutilise les connexions, détermineront la distribution du trafic. Les points de terminaison du lecteur suivent les changements de cluster en temps réel au fur et à mesure que des réplicas sont ajoutés ou supprimés. Vous pouvez placer ElastiCache les multiples répliques de lecture de votre cluster Redis OSS dans différentes zones de AWS disponibilité (AZ) afin de garantir la haute disponibilité des points de terminaison des lecteurs.

> **Note**
>
> Un point de terminaison du lecteur n'est pas un équilibreur de charge. Il s'agit d'un enregistrement DNS qui se résoudra à une adresse IP de l'un des nœuds de réplica en mode round robin.

Pour l'activité de lecture, les applications peuvent également se connecter à n'importe quel nœud du cluster. Contrairement au point de terminaison principal, les points de terminaison des nœuds sont résolus dans des points de terminaison spécifiques. En cas de modification dans votre cluster, comme l'ajout ou la suppression d'un réplica, vous devez mettre à jour les points de terminaison des nœuds de votre application.

Pour rechercher les points de terminaison d'un cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.

   L'écran des clusters apparaîtra avec une liste de clusters Valkey, Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé).
3. Pour trouver les points de terminaison du and/or lecteur principal du cluster, choisissez le nom du cluster (et non le bouton à gauche).

   Points de terminaison principaux et lecteurs pour un cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé)

   S'il y a un seul nœud dans le cluster, il n'y a aucun point de terminaison principal et vous pouvez passer à l'étape suivante.
4. Si le cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé) possède des nœuds de réplication, vous pouvez trouver les points de terminaison des nœuds de réplication du cluster en choisissant le nom du cluster, puis en choisissant l'onglet Nœuds.

   L'écran des nœuds s'affiche avec chaque nœud du cluster, principal et de réplica, répertorié avec son point de terminaison.

Node Name	Status	Current Role	Port	Endpoint
test-nc-001	available	primary	6379	amazonaws.com
test-nc-002	available	replica	6379	amazonaws.com
test-nc-003	available	replica	6379	amazonaws.com

Points de terminaison de nœuds pour un cluster Valkey, Valkey ou Redis OSS (mode cluster désactivé)

5. Pour copier un point de terminaison dans le presse-papiers :

   a. Un point de terminaison à la fois, recherchez le point de terminaison que vous souhaitez copier.

   b. Cliquez sur l'icône de copie située juste devant le point de terminaison.

   Le point de terminaison est maintenant copié dans votre presse-papiers. Pour en savoir plus sur l'utilisation du point de terminaison pour se connecter à un nœud, consultez Connexion aux nœuds Memcached.

Un point de terminaison principal Valkey, Valkey ou Redis OSS (mode cluster désactivé) ressemble à ce qui suit. Il existe une différence selon que le chiffrement en transit est activé ou pas.

Chiffrement en transit non activé

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port

redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Chiffrement en transit activé

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port

master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Recherche de points de terminaison pour un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Un cluster Valkey ou Redis OSS (mode cluster activé) possède un point de terminaison de configuration unique. En se connectant au point de terminaison de configuration, votre application est en mesure de découvrir les points de terminaison principal et de lecture pour chaque partition du cluster.

Pour trouver le point de terminaison d'un cluster Valkey ou Redis OSS (mode cluster activé)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.

   L'écran des clusters apparaîtra avec une liste de clusters. Choisissez le cluster auquel vous souhaitez vous connecter.
3. Pour rechercher le point de terminaison de la configuration du cluster, choisissez le nom du cluster (et non le bouton radio).
4. Le Configuration endpoint (point de terminaison de la configuration) est affiché sous Cluster details (Détails du cluster). Pour le copier, choisissez l'icône de copie à gauche du point de terminaison.

Trouver les points de terminaison d'un cluster (console) (Memcached)

Tous les points de terminaison Memcached sont des points de terminaison. read/write Pour vous connecter aux nœuds dans un cluster Memcached, votre application peut utiliser les points de terminaison de chaque nœud ou le point de terminaison de configuration du cluster avec la découverte automatique. Pour utiliser la découverte automatique, vous devez avoir recours à un client qui prend en charge la découverte automatique.

Lorsque vous utilisez la découverte automatique, votre application client se connecte à votre cluster Memcached en utilisant le point de terminaison de configuration. Lorsque vous redimensionnez votre cluster en ajoutant ou supprimant des nœuds, votre application « connaîtra » automatiquement tous les nœuds du cluster et sera en mesure de se connecter à l'un d'eux. Sans la fonction de découverte automatique, votre application devra faire cela ou vous devrez mettre à jour manuellement les points de terminaison de votre application après chaque ajout ou suppression d'un nœud.

Pour copier un point de terminaison, cliquez sur l'icône de copie située juste devant l'adresse du point de terminaison. Pour en savoir plus sur l'utilisation du point de terminaison pour se connecter à un nœud, consultez Connexion aux nœuds Memcached.

La configuration et les points de terminaison des nœuds sont très similaires. Les différences sont mises en évidence en gras comme suit.

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port   # configuration endpoint
 contains "cfg"
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port  # node endpoint for node 0001
```

> **Important**
>
> Si vous choisissez de créer un enregistrement CNAME pour votre point de terminaison de configuration Memcached, afin que votre client de découverte automatique reconnaisse l'enregistrement CNAME comme point de terminaison de configuration, vous devez inclure `.cfg.` dans le CNAME.

Recherche des points de terminaison (AWS CLI)

Pour Memcached, vous pouvez utiliser le for AWS CLI Amazon ElastiCache pour découvrir les points de terminaison des nœuds et des clusters.

Pour Redis OSS, vous pouvez utiliser le AWS CLI for Amazon ElastiCache pour découvrir les points de terminaison des nœuds, des clusters et également des groupes de réplication.

Rubriques

- Recherche des points de terminaison des nœuds et des clusters (AWS CLI)
- Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI

Recherche des points de terminaison des nœuds et des clusters (AWS CLI)

Vous pouvez utiliser le AWS CLI pour découvrir les points de terminaison d'un cluster et de ses nœuds à l'aide de la `describe-cache-clusters` commande. Pour les clusters Valkey ou Redis OSS, la commande renvoie le point de terminaison du cluster. Pour les clusters Memcached, la commande renvoie le point de terminaison de configuration. Si vous ajoutez le paramètre facultatif `--show-cache-node-info`, la commande renvoie également les points de terminaison de chaque nœud figurant dans le cluster.

Example

La commande suivante extrait le point de terminaison de configuration (`ConfigurationEndpoint`) et les points de terminaison des nœuds individuels (`Endpoint`) pour le cluster Memcached mycluster.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id mycluster \
    --show-cache-node-info
```

Pour Windows :

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id mycluster ^
    --show-cache-node-info
```

Le résultat de l'opération ci-dessus doit ressembler à cet exemple (format JSON).

```
{
   "CacheClusters": [
   {
       "Engine": "memcached",
       "CacheNodes": [
          {
             "CacheNodeId": "0001",
             "Endpoint": {
                "Port": 11211,
                "Address": "mycluster.amazonaws.com"
             },
                "CacheNodeStatus": "available",
                "ParameterGroupStatus": "in-sync",
                "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                "CustomerAvailabilityZone": "us-west-2b"
          },
          {
             "CacheNodeId": "0002",
             "Endpoint": {
                "Port": 11211,
                "Address": "mycluster.amazonaws.com"
             },
                "CacheNodeStatus": "available",
                "ParameterGroupStatus": "in-sync",
                "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                "CustomerAvailabilityZone": "us-west-2b"
          },
          {
                "CacheNodeId": "0003",
                "Endpoint": {
                   "Port": 11211,
                   "Address": "mycluster.amazonaws.com"
                },
                   "CacheNodeStatus": "available",
                   "ParameterGroupStatus": "in-sync",
                   "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
                   "CustomerAvailabilityZone": "us-west-2b"
          }
       ],
       "CacheParameterGroup": {
       "CacheNodeIdsToReboot": [],
       "CacheParameterGroupName": "default.memcached1.4",
```

```
        "ParameterApplyStatus": "in-sync"
             },
             "CacheClusterId": "mycluster",
             "PreferredAvailabilityZone": "us-west-2b",
             "ConfigurationEndpoint": {
                 "Port": 11211,
                 "Address": "mycluster.amazonaws.com"
             },
             "CacheSecurityGroups": [],
             "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
             "AutoMinorVersionUpgrade": true,
             "CacheClusterStatus": "available",
             "NumCacheNodes": 3,
             "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
             "CacheSubnetGroupName": "default",
             "EngineVersion": "1.4.24",
             "PendingModifiedValues": {},
             "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
             "CacheNodeType": "cache.m4.large",
              "DataTiering": "disabled"
         }
    ]
}
```

> **Important**
>
> Si vous choisissez de créer un enregistrement CNAME pour votre point de terminaison de configuration Memcached, afin que votre client de découverte automatique reconnaisse l'enregistrement CNAME comme point de terminaison de configuration, vous devez inclure `.cfg.` dans le CNAME. Par exemple, `mycluster.cfg.local` dans votre fichier php.ini pour le paramètre `session.save_path`.

Example

Pour Valkey et Redis OSS, la commande suivante récupère les informations de cluster pour le cluster à nœud unique mycluster.

> **Important**
>
> Le paramètre `--cache-cluster-id` peut être utilisé avec un identifiant de cluster Valkey ou Redis OSS (mode cluster désactivé) à nœud unique ou des identifiants de nœud spécifiques dans les groupes de réplication. Le `--cache-cluster-id` d'un groupe de réplication est une valeur à 4 chiffres telle que`0001`. S'il s'`--cache-cluster-id`agit de l'identifiant d'un cluster (nœud) dans un groupe de réplication, `replication-group-id` il est inclus dans la sortie.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-clusters \
    --cache-cluster-id redis-cluster \
    --show-cache-node-info
```

Pour Windows :

```
aws elasticache describe-cache-clusters ^
    --cache-cluster-id redis-cluster ^
    --show-cache-node-info
```

Le résultat de l'opération ci-dessus doit ressembler à cet exemple (format JSON).

```
{
    "CacheClusters": [
        {
            "CacheClusterStatus": "available",
            "SecurityGroups": [
                {
                    "SecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
            "CacheNodes": [
                {
                    "CustomerAvailabilityZone": "us-east-1b",
                    "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",
                    "CacheNodeStatus": "available",
                    "CacheNodeId": "0001",
                    "Endpoint": {
```

```
                        "Address": "redis-cluster.amazonaws.com",
                        "Port": 6379
                    },
                    "ParameterGroupStatus": "in-sync"
                }
            ],
            "AtRestEncryptionEnabled": false,
            "CacheClusterId": "redis-cluster",
            "TransitEncryptionEnabled": false,
            "CacheParameterGroup": {
                "ParameterApplyStatus": "in-sync",
                "CacheNodeIdsToReboot": [],
                "CacheParameterGroupName": "default.redis3.2"
            },
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-east-1b",
            "AutoMinorVersionUpgrade": true,
            "Engine": "redis",
            "AuthTokenEnabled": false,
            "PendingModifiedValues": {},
            "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
            "CacheSecurityGroups": [],
            "CacheSubnetGroupName": "default",
            "CacheNodeType": "cache.t2.small",
             "DataTiering": "disabled"
            "EngineVersion": "3.2.10",
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
            "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
        }
    ]
}
```

Pour de plus amples informations, consultez la rubrique describe-cache-clusters.

Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI

Vous pouvez utiliser le AWS CLI pour découvrir les points de terminaison d'un groupe de réplication et de ses clusters à l'aide de la `describe-replication-groups` commande. La commande renvoie le point de terminaison principal du groupe de réplication et une liste de tous les clusters (nœuds) figurant dans le groupe de réplication avec leurs points de terminaison et avec le point de terminaison du lecteur.

L'opération suivante extrait le point de terminaison principal et les points de terminaison du lecteur pour le groupe de réplication `myreplgroup`. Utilisez le point de terminaison principal pour toutes les opérations d'écriture.

```
aws elasticache describe-replication-groups \
    --replication-group-id myreplgroup
```

Pour Windows :

```
aws elasticache describe-replication-groups ^
    --replication-group-id myreplgroup
```

Le résultat de cette opération doit ressembler à cet exemple (format JSON).

```
{
   "ReplicationGroups": [
     {
       "Status": "available",
       "Description": "test",
       "NodeGroups": [
         {
            "Status": "available",
               "NodeGroupMembers": [
                  {
                     "CurrentRole": "primary",
                     "PreferredAvailabilityZone": "us-west-2a",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "myreplgroup-001.amazonaws.com"
                     },
                     "CacheClusterId": "myreplgroup-001"
                  },
                  {
                     "CurrentRole": "replica",
                     "PreferredAvailabilityZone": "us-west-2b",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "myreplgroup-002.amazonaws.com"
                     },
                     "CacheClusterId": "myreplgroup-002"
```

```
                    },
                    {
                       "CurrentRole": "replica",
                       "PreferredAvailabilityZone": "us-west-2c",
                       "CacheNodeId": "0001",
                       "ReadEndpoint": {
                          "Port": 6379,
                          "Address": "myreplgroup-003.amazonaws.com"
                       },
                       "CacheClusterId": "myreplgroup-003"
                    }
                 ],
                 "NodeGroupId": "0001",
                 "PrimaryEndpoint": {
                    "Port": 6379,
                    "Address": "myreplgroup.amazonaws.com"
                 },
                 "ReaderEndpoint": {
                    "Port": 6379,
                    "Address": "myreplgroup-ro.amazonaws.com"
                 }
              }
           ],
           "ReplicationGroupId": "myreplgroup",
           "AutomaticFailover": "enabled",
           "SnapshottingClusterId": "myreplgroup-002",
           "MemberClusters": [
              "myreplgroup-001",
              "myreplgroup-002",
              "myreplgroup-003"
           ],
           "PendingModifiedValues": {}
        }
     ]
}
```

Pour plus d'informations, consultez describe-replication-groups dans la Référence des commandes de l'AWS CLI .

Recherche de points de terminaison (ElastiCache API)

Pour Memcached, vous pouvez utiliser l' ElastiCache API Amazon pour découvrir les points de terminaison des nœuds et des clusters.

Pour Redis OSS, vous pouvez utiliser l' ElastiCache API Amazon pour découvrir les points de terminaison des nœuds, des clusters et des groupes de réplication.

Rubriques

- Recherche de points de terminaison pour les nœuds et les clusters (ElastiCache API)
- Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

Recherche de points de terminaison pour les nœuds et les clusters (ElastiCache API)

Vous pouvez utiliser l' ElastiCache API pour découvrir les points de terminaison d'un cluster et de ses nœuds grâce à l'`DescribeCacheClusters`action. Pour les clusters Valkey ou Redis OSS, la commande renvoie le point de terminaison du cluster. Pour les clusters Memcached, la commande renvoie le point de terminaison de configuration. Si vous ajoutez le paramètre facultatif `ShowCacheNodeInfo`, l'action renvoie également les points de terminaison de chaque nœud figurant dans le cluster.

Example

Pour Memcached, la commande suivante récupère le point de terminaison de configuration (*`ConfigurationEndpoint`*) et les points de terminaison des nœuds individuels (*`Endpoint`*) pour le cluster Memcached mycluster.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=mycluster
    &ShowCacheNodeInfo=true
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

> **Important**
>
> Si vous choisissez de créer un enregistrement CNAME pour votre point de terminaison de configuration Memcached, afin que votre client de découverte automatique reconnaisse l'enregistrement CNAME comme point de terminaison de configuration, vous devez inclure `.cfg.` dans le CNAME. Par exemple, `mycluster.cfg.local` dans votre fichier php.ini pour le paramètre `session.save_path`.

Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

Vous pouvez utiliser l' ElastiCache API pour découvrir les points de terminaison d'un groupe de réplication et de ses clusters grâce à l'`DescribeReplicationGroups`action. L'action renvoie le point de terminaison principal du groupe de réplication et une liste de tous les clusters figurant dans le groupe de réplication avec le point de terminaison du lecteur.

L'opération suivante permet de récupérer le point de terminaison principal (PrimaryEndpoint), le point de terminaison du lecteur (ReaderEndpoint) et les points de terminaison des nœuds individuels (ReadEndpoint) pour le groupe de réplication. `myreplgroup` Utilisez le point de terminaison principal pour toutes les opérations d'écriture.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=myreplgroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

Pour de plus amples informations, veuillez consulter DescribeReplicationGroups.

Utilisation de fragments dans ElastiCache

Un shard (API/CLI : groupe de nœuds) est une collection de un à six nœuds Valkey ou ElastiCache Redis OSS. Un cluster Valkey ou Redis OSS (mode cluster désactivé) ne comportera jamais plus d'une partition. Avec les partitions, vous pouvez séparer les grandes bases de données en parties plus petites, plus rapides et plus faciles à gérer, appelées fragments de données. Cela peut améliorer l'efficacité de la base de données en répartissant les opérations entre plusieurs sections distinctes.

L'utilisation de partitions peut offrir de nombreux avantages, notamment une amélioration des performances, de l'évolutivité et de la rentabilité.

Vous pouvez créer un cluster contenant un nombre de partitions supérieur et un nombre de réplicas inférieur, qui conduisent à un nombre total de 90 nœuds par cluster. Cette configuration de cluster peut contenir de 90 partitions avec 0 réplica à 15 partitions avec 5 réplicas, ce qui correspond au nombre maximal de réplicas autorisé. Les données du cluster sont partitionnées entre les partitions du cluster. S'il existe plusieurs nœuds dans une partition, la partition implémente la réplication, l'un des nœuds étant le nœud read/write principal et les autres nœuds étant des nœuds de réplication en lecture seule.

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur est Valkey 7.2 ou supérieure, ou Redis OSS 5.0.6 à 7.1. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Lorsque vous créez un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de la ElastiCache console, vous spécifiez le nombre de partitions dans le cluster et le nombre de nœuds dans les partitions. Pour de plus amples informations, veuillez consulter Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console). Si vous utilisez l' ElastiCache API ou AWS CLI pour créer un cluster (appelé groupe de réplication dans le groupe de nœuds API/CLI), you can configure the number of nodes in a shard (API/CLI :) de manière indépendante. Pour plus d'informations, consultez les ressources suivantes :

- API : CreateReplicationGroup
- CLI : create-replication-group

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

Configurations de partage Valkey ou Redis OSS

Pour plus d'informations, consultez Repartage hors ligne pour Valkey ou Redis OSS (mode cluster activé) et Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé).

Recherche de l'ID d'une partition

Vous pouvez trouver l'ID d'une partition à l'aide de l'API AWS Management Console, de AWS CLI ou de l' ElastiCache API.

À l'aide du AWS Management Console

Rubriques

- Pour Valkey ou Redis OSS (mode cluster désactivé)
- Pour Valkey ou Redis OSS (mode cluster activé)

Pour Valkey ou Redis OSS (mode cluster désactivé)

Les partitions du groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) le sont toujours IDs . `0001`

Pour Valkey ou Redis OSS (mode cluster activé)

La procédure suivante utilise le AWS Management Console pour rechercher l'ID de partition d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé).

Pour trouver l'ID de partition dans un groupe de réplication Valkey ou Redis OSS (mode cluster activé)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS, puis le nom du groupe de réplication Valkey ou Redis OSS (mode cluster activé) pour lequel vous souhaitez rechercher le shard. IDs
3. Dans la colonne Shard Name (Nom de la partition), l'ID de la partition correspond aux quatre derniers chiffres du nom de la partition.

À l'aide du AWS CLI

Pour trouver les identifiants de partition (groupe de nœuds) pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé), utilisez l' AWS CLI opération `describe-replication-groups` avec le paramètre facultatif suivant.

- **`--replication-group-id`** – paramètre facultatif qui, lorsqu'il est utilisé, limite la sortie aux informations correspondant au groupe de réplication spécifié. Si ce paramètre n'est pas utilisé, vous obtenez les informations relatives à 100 groupes de réplication maximum.

Example

Cette commande renvoie les informations relatives à `sample-repl-group`.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-replication-groups \
    --replication-group-id sample-repl-group
```

Pour Windows :

```
aws elasticache describe-replication-groups ^
    --replication-group-id sample-repl-group
```

Le résultat de cette commande ressemble à ceci. Les identifiants de shard (groupe de nœuds) sont *`highlighted`* là pour faciliter leur recherche.

```
{
```

```
    "ReplicationGroups": [
        {
            "Status": "available",
            "Description": "2 shards, 2 nodes (1 + 1 replica)",
            "NodeGroups": [
                {
                    "Status": "available",
                    "Slots": "0-8191",
                    "NodeGroupId": "0001",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2c",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0001-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0001-002"
                        }
                    ]
                },
                {
                    "Status": "available",
                    "Slots": "8192-16383",
                    "NodeGroupId": "0002",
                    "NodeGroupMembers": [
                        {
                            "PreferredAvailabilityZone": "us-west-2b",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0002-001"
                        },
                        {
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CacheNodeId": "0001",
                            "CacheClusterId": "sample-repl-group-0002-002"
                        }
                    ]
                }
            ],
            "ConfigurationEndpoint": {
                "Port": 6379,
                "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
```

```
            },
            "ClusterEnabled": true,
            "ReplicationGroupId": "sample-repl-group",
            "SnapshotRetentionLimit": 1,
            "AutomaticFailover": "enabled",
            "SnapshotWindow": "13:00-14:00",
            "MemberClusters": [
                "sample-repl-group-0001-001",
                "sample-repl-group-0001-002",
                "sample-repl-group-0002-001",
                "sample-repl-group-0002-002"
            ],
            "CacheNodeType": "cache.m3.medium",
            "DataTiering": "disabled",
            "PendingModifiedValues": {}
        }
    ]
}
```

Utilisation de l' ElastiCache API

Pour trouver les identifiants de partition (groupe de nœuds) pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé), utilisez l' AWS CLI opération `describe-replication-groups` avec le paramètre facultatif suivant.

- **`ReplicationGroupId`** : paramètre facultatif qui, lorsqu'il est utilisé, limite la sortie aux informations correspondant au groupe de réplication spécifié. Si ce paramètre est omis, les détails d'un maximum de groupes de *`xxx`* réplication sont renvoyés.

Example

Cette commande renvoie les informations relatives à `sample-repl-group`.

Pour Linux, macOS ou Unix :

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroup
   &ReplicationGroupId=sample-repl-group
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Comparaison des caches auto-conçus par Valkey, Memcached et Redis OSS

Amazon ElastiCache prend en charge les moteurs de cache Valkey, Memcached et Redis OSS. Chaque moteur offre certains avantages. Utilisez les informations dans cette rubrique pour vous aider à choisir le moteur et la version que répond le mieux à vos besoins.

> **Important**
>
> Après avoir créé un cache, un cluster conçu par vos soins ou un groupe de réplication, vous pouvez effectuer une mise à niveau vers une version du moteur plus récente, mais vous ne pouvez pas rétrograder vers une ancienne version du moteur. Si vous souhaitez utiliser une ancienne version du moteur, vous devez supprimer le cache, le cluster conçu par vous-même ou le groupe de réplication existant et le créer à nouveau avec la version précédente du moteur.

En apparence, les moteurs de ressemblent. Chacun de ces moteurs est une banque de clés en mémoire-valeurs. Toutefois, en pratique, il existe des différences significatives.

Choisissez Memcached si les éléments suivants s'appliquent à votre situation :

- Vous avez besoin du modèle le plus simple possible.
- Vous avez besoin d'exécuter des nœuds de grande taille avec plusieurs cœurs ou threads.
- Vous devez pouvoir le faire évoluer en ajoutant et en supprimant des nœuds au fur et à mesure des fluctuations du système.
- Vous avez besoin d'objets du cache.

Choisissez Valkey ou Redis OSS avec ElastiCache si les conditions suivantes s'appliquent à vous :

- ElastiCache version 7.2 pour Valkey ou version 7.0 (améliorée) pour Redis OSS

  Vous souhaitez utiliser les améliorations apportées à Functions, Sharded Pub/Sub ou ACL. Pour plus d'informations, consultez Redis OSS version 7.0 (améliorée).

- ElastiCache version 6.2 (améliorée) pour Redis OSS

  Vous souhaitez pouvoir hiérarchiser les données entre la mémoire et le SSD à l'aide du type de nœud r6gd. Pour plus d'informations, consultez la rubrique Hiérarchisation des données.

- ElastiCache version 6.0 (améliorée) pour Redis OSS

  Vous souhaitez authentifier les utilisateurs avec le contrôle d'accès basé sur les rôles.

  Pour plus d'informations, consultez Redis OSS version 6.0 (améliorée).

- ElastiCache version 5.0.0 (améliorée) pour Redis OSS

  Vous souhaitez utiliser les flux Redis OSS, une structure de données de journal qui permet aux producteurs d'ajouter de nouveaux éléments en temps réel et aux consommateurs de consommer des messages de manière bloquante ou non bloquante.

  Pour plus d'informations, consultez Redis OSS version 5.0.0 (améliorée).

- ElastiCache version 4.0.10 (améliorée) pour Redis OSS

  Supporte à la fois le chiffrement et l'ajout ou la suppression dynamiques de fragments de votre cluster Valkey ou Redis OSS (mode cluster activé).

  Pour plus d'informations, consultez Redis OSS version 4.0.10 (améliorée).

Les versions suivantes sont obsolètes, ont atteint ou atteindront bientôt leur fin de vie.

- ElastiCache version 3.2.10 (améliorée) pour Redis OSS

  Permet d'ajouter ou de supprimer dynamiquement des fragments de votre cluster Valkey ou Redis OSS (mode cluster activé).

  > **Important**
  >
  > Actuellement, la ElastiCache version 3.2.10 pour Redis OSS ne prend pas en charge le chiffrement.

  Pour plus d'informations, consultez les ressources suivantes :

  - Version 3.2.10 de Redis OSS (améliorée)

- Meilleures pratiques en matière de repartage en ligne pour Redis OSS. Pour plus d'informations, consultez ce qui suit :
  - Bonnes pratiques : repartitionnement en ligne
  - Repartage en ligne et rééquilibrage des partitions pour Valkey ou Redis OSS (mode cluster activé)
- Pour plus d'informations sur le dimensionnement des clusters Redis OSS, consultez la section Mise à l'échelle.

- ElastiCache version 3.2.6 (améliorée) pour Redis OSS

  Si vous avez besoin des fonctionnalités des versions antérieures de Redis OSS ainsi que des fonctionnalités suivantes, choisissez 3.2.6 :

  - Chiffrement en transit. Pour plus d'informations, consultez Amazon ElastiCache for Redis OSS In-Transit Encryption.
  - Chiffrement au repos. Pour plus d'informations, consultez Amazon ElastiCache for Redis OSS At-Rest Encryption.
- ElastiCache (Mode cluster activé) version 3.2.4 pour Redis OSS

  Si vous avez besoin des fonctionnalités de la version 2.8.x ainsi que des fonctionnalités suivantes, choisissez la version 3.2.4 (mode clusterisé) :

  - Vous avez besoin de partitionner vos données sur deux à 500 groupes de nœuds (mode cluster uniquement).
  - Vous avez besoin d'indexation géospatiale (mode cluster ou non-cluster).
  - Vous n'avez pas besoin de prendre en charge plusieurs bases de données.
- ElastiCache (mode non clusterisé) 2.8.x et 3.2.4 (amélioré) pour Redis OSS

  Si les conditions suivantes s'appliquent à vous, 2.8.x ou 3.2.4 (mode non clusterisé) :

  - Vous avez besoin de types de données complexes, telles que des chaînes, des données de hachage, des listes, des ensembles triés et des bitmaps.
  - Vous avez besoin de trier ou de classer des ensembles de données en mémoire.
  - Vous avez besoin que votre banque de clés soit maintenue.
  - Vous avec besoin de répliquer les données du réplica principal vers un ou plusieurs réplicas en lecture pour des applications de lecture intensive.

- Vous avez besoin de procéder à un basculement automatiquement en cas de défaillance de votre nœud principal.
- Vous avez besoin de publier et de vous abonner à des fonctionnalités (pub/sub), pour informer les clients sur des événements sur le serveur.
- Vous avez besoin de fonctionnalités de sauvegarde et de restauration pour les clusters conçus par vos soins ainsi que pour les caches sans serveur.
- Vous avez besoin de prendre en charge plusieurs bases de données.

Résumé de la comparaison entre Memcached, Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé)

	Memcached	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
Versions du moteur+	1.4.5 et versions ultérieures	4.0.10 et ultérieures	4.0.10 et ultérieures
Types de données	Simplicité	2.8.x - Complexe * Complexe	2.8.x et version ultérieure – Complexe
Partitionnement de données	Oui	Non	Oui
Le cluster est modifiable	Oui	Oui	Windows 3.2.10 et version ultérieure – Limité
Repartitionnement en ligne	Non	Non	3.2.10 et ultérieures
Chiffrement	in-transit 1.6.12 et versions ultérieures	4.0.10 et ultérieures	4.0.10 et ultérieures
Mise à niveau des données	Non	6.2 et versions ultérieures	6.2 et versions ultérieures
Certifications de conformité			
Certification de conformité			
FedRAMP	Oui - 1.6.12 et ultérieure	4.0.10 et ultérieures	4.0.10 et ultérieures
HIPAA	Oui - 1.6.12 et ultérieure	4.0.10 et ultérieures	4.0.10 et ultérieures
PCI DSS	Oui	4.0.10 et ultérieures	4.0.10 et ultérieures

	Memcached	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
Multithread	Oui	Non	Non
Mise à niveau du type de nœud	Non	Oui	Oui
Mise à niveau du moteur	Oui	Oui	Oui
Haute disponibilité (réplication)	Non	Oui	Oui
Basculement automatique	Non	Facultatif	Obligatoire
Fonctionnalités Pub/Sub	Non	Oui	Oui
Ensembles triés	Non	Oui	Oui
Sauvegarde et restauration	Pour Memcached sans serveur uniquement, pas pour les clusters Memcached conçus par vos soins	Oui	Oui
Indexation géospatiale	Non	4.0.10 et ultérieures	Oui

Remarques:

chaîne, objets (tels que les bases de données)

* chaînes, ensembles, ensembles triés, listes, hachages, bitmaps, hyperloglog

chaîne, ensembles, ensembles triés, listes, hachages, bitmaps, hyperloglog, index géospatiaux

	Memcached	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)

+ Exclut les versions obsolètes, en fin de vie ou sur le point de l'être.

Une fois que vous avez choisi le moteur de votre cluster, nous vous conseillons d'utiliser la version la plus récente de ce moteur. Pour de plus amples informations, veuillez consulter Types de nœuds pris en charge.

Migration en ligne pour Valkey ou Redis OSS

En utilisant la migration en ligne, vous pouvez migrer vos données depuis votre logiciel libre Valkey ou Redis OSS auto-hébergé sur Amazon vers Amazon. EC2 ElastiCache

Cela fait référence à la migration d'une instance auto-hébergée vers le ElastiCache service. Pour plus d'informations sur la mise à niveau de Redis OSS vers Valkey, voir. ElastiCache Mise à niveau des versions du moteur, y compris les mises à niveau entre

> **Note**
>
> La migration en ligne n'est pas prise en charge vers les caches ou clusters ElastiCache sans serveur exécutés sur le type de nœud r6gd.

Présentation

Pour migrer vos données depuis le logiciel open source Valkey ou Redis OSS exécuté sur Amazon vers EC2 Amazon, vous devez ElastiCache disposer d'un déploiement Amazon existant ou récemment créé. ElastiCache Le déploiement doit posséder une configuration prête pour la migration. Il doit également être conforme à la configuration que vous voulez, y compris les attributs que sont notamment le type d'instance, le nombre de partitions et le nombre de réplicas.

La migration en ligne est conçue pour la migration de données à partir de Valkey ou Redis OSS open source auto-hébergés sur Amazon EC2 vers ElastiCache, et non pour déplacer des données entre des clusters. ElastiCache

> **Important**
>
> Il est vivement recommandé de lire les sections suivantes intégralement avant de commencer le processus de migration.

La migration commence lorsque vous appelez l'opération d'API `StartMigration` ou la commande AWS CLI . Lors de la migration de clusters désactivés en mode cluster Valkey ou Redis OSS, le processus de migration fait du nœud principal du cluster Valkey ou Redis OSS une réplique de votre ElastiCache Valkey ou Redis OSS principal source. Lors de la migration de clusters compatibles avec Valkey ou Redis OSS en mode cluster, le processus de migration fait du nœud principal de chaque partition une réplique de la ElastiCache partition correspondante de votre cluster source possédant les mêmes emplacements.

Une fois que les modifications côté client sont prêtes, appelez l'opération d'API `CompleteMigration`. Cette opération d'API favorise votre ElastiCache déploiement vers votre déploiement principal de Valkey ou Redis OSS avec des nœuds principaux et répliques (le cas échéant). Vous pouvez maintenant rediriger votre application client pour commencer à écrire les données sur ElastiCache. Tout au long de la migration, vous pouvez vérifier l'état de la réplication en exécutant la commande valkey-cli INFO sur vos nœuds Valkey et sur les nœuds principaux. ElastiCache

Étapes de la migration

Les rubriques suivantes décrivent le processus à suivre pour migrer vos données :

- Préparation de votre source et de votre cible pour la migration
- Test de la migration des données
- Démarrage d'une migration
- Vérification du déroulement de la migration des données
- Achèvement de la migration des données

Préparation de votre source et de votre cible pour la migration

En suivant ces étapes, vous pouvez vous préparer à migrer vos données d'une source Valkey ou Redis auto-hébergée EC2 vers ElastiCache, ou depuis un cluster Redis OSS vers un cluster Valkey. ElastiCache

Cela fait référence à la migration d'une instance auto-hébergée vers le ElastiCache service. Pour plus d'informations sur la mise à niveau de Redis OSS vers Valkey, voir. ElastiCache Mise à niveau des versions du moteur, y compris les mises à niveau entre

Vous devez vous assurer que les quatre conditions préalables mentionnées ci-dessous sont satisfaites avant de démarrer la migration depuis la ElastiCache console, l'API ou la AWS CLI.

Pour préparer vos nœuds Valkey ou Redis OSS source et cible à la migration

1. Identifiez le ElastiCache déploiement cible et assurez-vous que vous pouvez y faire migrer les données.

   Un ElastiCache déploiement existant ou nouvellement créé doit répondre aux exigences suivantes en matière de migration :

   - Il utilise Valkey ou Redis OSS 5.0.6 ou supérieur.
   - Le chiffrement en transit n'est pas activé.
   - Il a Multi-AZ activé.
   - Il dispose de suffisamment de mémoire pour contenir les données de votre cluster Valkey ou Redis OSS. Pour configurer les paramètres de mémoire réservés appropriés, consultez Gestion de la mémoire réservée pour Valkey et Redis OSS.
   - Si le mode cluster est désactivé, vous pouvez migrer directement depuis les versions 2.8.21 de Valkey ou Redis OSS vers les versions 5.0.6 de Valkey ou Redis OSS si vous utilisez la CLI ou les versions 5.0.6 de Valkey ou Redis OSS à l'aide de l'interface de ligne de commande ou de la console. Si le mode cluster est activé, vous pouvez migrer directement depuis n'importe quelle version de Valkey ou Redis OSS activée en mode cluster vers Redis OSS version 5.0.6 ultérieure, si vous utilisez la CLI ou les versions 5.0.6 ultérieures de Redis OSS à l'aide de la CLI ou de la console.
   - Nombre identique de partitions dans la source et la cible.
   - Il ne fait pas partie d'un entrepôt de données global.
   - La hiérarchisation des données est désactivée.
2. Assurez-vous que les configurations de votre logiciel open source Valkey ou Redis OSS et le ElastiCache déploiement sont compatibles.

   Au minimum, tous les éléments suivants du ElastiCache déploiement cible doivent être compatibles avec votre configuration Valkey ou Redis OSS pour la réplication :

- AUTH ne doit pas être activé sur votre cluster.
- La configuration `protected-mode` doit être définie surno.
- Si vous avez une `bind` configuration dans votre configuration Valkey ou Redis OSS, elle doit être mise à jour pour autoriser les demandes provenant des ElastiCache nœuds.
- Le nombre de bases de données logiques doit être le même sur le ElastiCache nœud et sur votre cluster Valkey ou Redis OSS. Cette valeur est définie `databases` à l'aide de la configuration Valkey ou Redis OSS.
- Les commandes Valkey ou Redis OSS qui modifient les données ne doivent pas être renommées pour permettre la réplication des données. Par exemple`sync`,,, `psyncinfo`, `config` et. `command` `cluster`
- Pour répliquer les données de votre cluster Valkey ou Redis OSS vers ElastiCache, assurez-vous que le processeur et la mémoire sont suffisants pour gérer cette charge supplémentaire. Cette charge provient du fichier RDB créé par votre cluster Valkey ou Redis OSS et transféré sur le réseau vers le nœud. ElastiCache
- Toutes les instances Valkey ou Redis OSS du cluster source doivent s'exécuter sur le même port.

3. Assurez-vous que vos instances peuvent se connecter ElastiCache en procédant comme suit :

   - Vérifiez que l'adresse IP de chaque instance est privée.
   - Attribuez ou créez le ElastiCache déploiement dans le même cloud privé virtuel (VPC) que votre Valkey ou Redis OSS sur votre instance (recommandé).
   - S' VPCs ils sont différents, configurez le peering VPC pour autoriser l'accès entre les nœuds. Pour de plus amples informations sur l'appairage de VPC, consulter Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC.
   - Le groupe de sécurité attaché à vos instances Valkey ou Redis OSS doit autoriser le trafic entrant en provenance des nœuds. ElastiCache

4. Assurez-vous que votre application peut diriger le trafic vers les ElastiCache nœuds une fois la migration des données terminée. Pour de plus amples informations, veuillez consulter Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC.

Test de la migration des données

Une fois toutes les conditions requises remplies, vous pouvez valider la configuration de la migration à l' AWS Management Console aide de l' ElastiCache API ou AWS CLI. L'exemple suivant montre comment utiliser la CLI.

Testez la migration en appelant la commande `test-migration` avec les paramètres suivants :

- `--replication-group-id` – ID du groupe de réplication vers lequel les données doivent être migrées.
- `--customer-node-endpoint-list` – Liste des points de terminaison à partir desquels les données doivent être migrées. La liste ne doit comporter qu'un seul élément.

Voici un exemple avec utilisation de la CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-
endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validera la configuration de la migration sans migration de données réelle.

Démarrage d'une migration

Une fois toutes les conditions requises remplies, vous pouvez commencer la migration des données à l' AWS Management Console aide de l' ElastiCache API ou AWS CLI. Si le mode cluster est activé et que la migration des emplacements diffère, un repartitionnement est effectué avant la migration en direct. L'exemple suivant montre comment utiliser la CLI.

> **Note**
>
> Nous avons recommandé d'utiliser l'API `TestMigration` pour valider la configuration de la migration. Mais cela est strictement facultatif.

Commencez la migration en appelant la commande `start-migration` avec les paramètres suivants :

- `--replication-group-id`— Identifiant du groupe de ElastiCache réplication cible
- `--customer-node-endpoint-list`— Une liste de points de terminaison avec des adresses DNS ou IP et le port sur lequel votre cluster source Valkey ou Redis OSS est exécuté. La liste ne

peut accepter qu'un seul élément à la fois pour le mode cluster désactivé et pour le mode cluster activé. Si vous avez activé la réplication en chaîne, le point de terminaison peut pointer vers une réplique au lieu du nœud principal de votre cluster Valkey ou Redis OSS.

Voici un exemple avec utilisation de la CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-
endpoint-list "Address='10.0.0.241',Port=6379"
```

Lorsque vous exécutez cette commande, le nœud ElastiCache principal (dans chaque partition) se configure pour devenir une réplique de votre instance Valkey ou Redis OSS (dans la partition correspondante possédant les mêmes emplacements dans Redis activé par le cluster). L'état du ElastiCache cluster passe à la migration et les données commencent à migrer depuis votre instance Valkey ou Redis OSS vers le nœud principal. ElastiCache En fonction de la taille des données et de la charge sur votre instance Valkey ou Redis OSS, la migration peut prendre un certain temps. Vous pouvez vérifier la progression de la migration en exécutant la commande valkey-cli INFO sur votre instance Valkey et votre nœud principal. ElastiCache

Une fois la réplication réussie, toutes les écritures sur vos instances Valkey ou Redis OSS se propagent vers le cluster. ElastiCache Vous pouvez utiliser des ElastiCache nœuds pour les lectures. Cependant, vous ne pouvez pas écrire sur le cluster ElastiCache. Si d'autres nœuds de réplication sont connectés à un nœud ElastiCache principal, ces nœuds de réplication continuent à se répliquer à partir du nœud ElastiCache principal. De cette façon, toutes les données de votre cluster Valkey ou Redis OSS sont répliquées sur tous les nœuds du cluster. ElastiCache

Si un nœud ElastiCache principal ne peut pas devenir une réplique de votre instance Valkey ou Redis OSS, il réessaie plusieurs fois avant de redevenir principal. L'état du cluster ElastiCache devient alors available (disponible), et un événement de groupe de réplication concernant l'échec du lancement de la migration est envoyé. Pour résoudre un tel problème, vérifiez ce qui suit :

- Observez l'événement de groupe de réplication. Utilisez toute information spécifique de l'événement pour corriger l'échec de la migration.
- Si l'événement ne fournit pas d'informations spécifiques, assurez-vous d'avoir suivi les instructions dans Préparation de votre source et de votre cible pour la migration.
- Assurez-vous que la configuration de routage de votre VPC et de vos sous-réseaux autorise le trafic entre les ElastiCache nœuds et vos instances Valkey ou Redis OSS.

- Assurez-vous que le groupe de sécurité attaché à vos instances Valkey ou Redis OSS autorise le trafic entrant provenant des ElastiCache nœuds.
- Consultez les journaux Valkey ou Redis OSS de vos instances pour plus d'informations sur les défaillances spécifiques à la réplication.

Vérification du déroulement de la migration des données

Une fois la migration des données commencée, vous pouvez procéder comme suit pour suivre son déroulement :

- Vérifiez que Valkey ou `master_link_status` Redis OSS figure up dans la `INFO` commande sur le ou les nœuds ElastiCache principaux. Vous pouvez également trouver ces informations dans la ElastiCache console. Sélectionnez le cluster et, sous CloudWatch Metrics, observez Primary Link Health Status. Une fois que la valeur a atteint 1, les données sont synchronisées.
- Vous pouvez vérifier que la ElastiCache réplique possède un état en ligne en exécutant la `INFO` commande sur vos instances Valkey ou Redis OSS. Cela permet également d'obtenir des informations sur le retard de réplication.
- Vérifiez que la mémoire tampon de sortie du client est faible en utilisant la commande CLIENT LIST sur vos instances Valkey ou Redis OSS.

Une fois la migration des données terminée, les données sont synchronisées avec toutes les nouvelles écritures arrivant sur le ou les nœuds principaux de votre cluster Valkey ou Redis OSS.

Achèvement de la migration des données

Lorsque vous êtes prêt à passer au ElastiCache cluster, utilisez la commande `complete-migration` CLI avec les paramètres suivants :

- `--replication-group-id` – identifiant du groupe de réplication.
- `--force` – une valeur qui force la migration à s'arrêter sans s'assurer que les données sont synchronisées.

Voici un exemple.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

Lorsque vous exécutez cette commande, le nœud ElastiCache principal (dans chaque partition) arrête de se répliquer depuis votre instance Valkey ou Redis OSS et la transforme en nœud principal. Cette promotion se termine généralement en quelques minutes. Pour confirmer la promotion vers primaire, vérifiez l'événement. `Complete Migration successful for test-cluster`. À ce stade, vous pouvez orienter votre application vers l' ElastiCache écriture et la lecture. ElastiCache l'état du cluster devrait passer de la migration à la disponibilité.

Si la promotion vers le serveur principal échoue, le nœud ElastiCache principal continue de se répliquer depuis votre instance Valkey ou Redis OSS. Le ElastiCache cluster est toujours en état de migration et un message d'événement du groupe de réplication concernant l'échec est envoyé. Pour corriger cet échec, vérifiez les points suivants :

- Vérifiez l'événement de groupe de réplication. Utilisez les informations spécifiques de l'événement pour corriger l'échec.
- Il se peut que vous receviez un message d'événement à propos de données non synchronisées. Si tel est le cas, assurez-vous que le serveur ElastiCache principal peut se répliquer à partir de votre instance Valkey ou Redis OSS et que les deux sont synchronisés. Si vous voulez toujours arrêter la migration, vous pouvez exécuter la commande précédente avec l'option `–force`.
- Un message d'événement peut s'afficher si l'un des ElastiCache nœuds est en cours de remplacement. Vous pouvez relancer l'étape de migration complète une fois le remplacement terminé.

Effectuer la migration des données en ligne à l'aide de la console

Vous pouvez utiliser le AWS Management Console pour migrer vos données de votre cluster vers votre cluster Valkey ou Redis OSS.

Effectuer la migration des données en ligne à l'aide de la console

1. Connectez-vous à la console et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Créez un nouveau cluster Valkey ou Redis OSS ou choisissez un cluster existant. Vérifiez que le cluster satisafait aux exigences suivantes :
    - La version de votre moteur doit être Valkey 7.2 ou supérieure, ou Redis OSS 5.0.6 ou supérieure.
    - AUTH ne doit pas être activé sur votre cluster.

- La configuration `protected-mode` doit être définie surno.
- Si vous avez une `bind` configuration dans votre configuration Valkey ou Redis OSS, elle doit être mise à jour pour autoriser les demandes provenant des ElastiCache nœuds.
- Le nombre de bases de données doit être le même entre le ElastiCache nœud et votre cluster Valkey ou Redis OSS. Cette valeur est définie `databases` dans la configuration du moteur.
- Les commandes Valkey ou Redis OSS qui modifient les données ne doivent pas être renommées pour permettre à la réplication des données de réussir.
- Pour répliquer les données de votre cluster Valkey ou Redis OSS vers ElastiCache, assurez-vous que le processeur et la mémoire sont suffisants pour gérer cette charge supplémentaire. Cette charge provient du fichier RDB créé par votre cluster Valkey ou Redis OSS et transféré sur le réseau vers le nœud. ElastiCache
- L'éta du cluster est available (disponible).

3. Une fois votre cluster sélectionné, sélectionnez Migrate Data from Endpoint (Migrer les données à partir du point de terminaison pour les Actions.
4. Dans la boîte de dialogue Migrer les données depuis le point de terminaison, entrez l'adresse IP et le port sur lesquels votre cluster Valkey ou Redis OSS est disponible.

   > **Important**
   >
   > L'adresse IP doit être exacte. Si vous saisissez une adresse erronée, la migration échoue.

5. Choisissez Start Migration (Lancer la migration).

   Lorsque le cluster commence la migration, il passe à l'état Mofiying (Modification) puis à l'état Migrating (Migration).
6. Surveillez la progression de la migration en choisissant Events (Événements) dans le volet de navigation.

A tout moment au cours du processus de migration, vous pouvez l'arrêter. Pour ce faire, sélectionnez votre cluster et Stop Data Migration (Arrêter la migration des données) pour les Actions. Le cluster passe alors à l'état Available (Disponible).

Si la migration réussit, le cluster passe à l'état Available (Disponible) et le journal des événements affiche ce qui suit :

```
Migration operation succeeded for replication group ElastiCacheClusterName.
```

Si la migration échoue, le cluster passe à l'état Available (Disponible) et le journal des événements affiche ce qui suit :

```
Migration operation failed for replication group ElastiCacheClusterName.
```

Choix des régions et des zones de disponibilité pour ElastiCache

Vous pouvez apporter une évolutivité et une fiabilité supplémentaires à vos ElastiCache clusters en désignant des régions et des zones de disponibilité à l'aide du point de terminaison correspondant.

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 ElastiCache activité 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.

Régions et zones de disponibilité

Rubriques

- Considérations relatives à la zone de disponibilité avec Memcached
- Localisation de vos nœuds
- Régions et terminaux pris en charge
- Utilisation de zones locales avec ElastiCache
- Utiliser Outposts avec ElastiCache

Considérations relatives à la zone de disponibilité avec Memcached

La distribution de vos nœuds Memcached sur plusieurs zones de disponibilité au sein d'une région permet de vous protéger contre l'impact d'un échec catastrophique, telle qu'une panne de courant dans une zone de disponibilité.

Mise en cache sans serveur

ElastiCache la mise en cache sans serveur crée un cache hautement disponible qui couvre plusieurs zones de disponibilité. Vous pouvez spécifier des sous-réseaux provenant de différentes zones de disponibilité et du même VPC lorsque vous créez votre cluster sans serveur ElastiCache ou choisir automatiquement des sous-réseaux à partir de votre VPC par défaut.

Conception de votre propre cluster ElastiCache pour Memcached

Un cluster Memcached peut avoir jusqu'à 300 nœuds. Lorsque vous créez ou ajoutez des nœuds à votre cluster Memcached, vous pouvez spécifier une seule zone de disponibilité pour tous vos nœuds, autoriser le choix ElastiCache d'une seule zone de disponibilité pour tous vos nœuds, spécifier les zones de disponibilité pour chaque nœud ou autoriser ElastiCache le choix d'une zone de disponibilité pour chaque nœud. De nouveaux nœuds peuvent être créés dans différentes zones de disponibilité lorsque vous les ajoutez à un cluster Memcached existant. Une fois qu'un nœud de cache est créé, sa zone de disponibilité ne peut pas être modifiée.

Si vous souhaitez que les nœuds d'un cluster d'une seule zone de disponibilité soient répartis sur plusieurs zones de disponibilité, ElastiCache vous pouvez créer de nouveaux nœuds dans les différentes zones de disponibilité. Vous pouvez ensuite supprimer tout ou partie des nœuds de cache d'origine. Nous recommandons cette approche.

Pour migrer des nœuds Memcached d'une seule zone de disponibilité vers plusieurs zones de disponibilité

1. Modifiez votre cluster en créant des nœuds de cache dans les zones de disponibilité où vous les souhaitez. Dans votre demande, effectuez les opérations suivantes :

   - Définissez `AZMode` (CLI :`- -az-mode`) sur `cross-az`.
   - Définissez `NumCacheNodes` (CLI :`- -num-cache-nodes`) sur le nombre de nœuds de cache actuellement actifs ainsi que sur le nombre de nouveaux nœuds de cache que vous souhaitez créer.
   - Définissez `NewAvailabilityZones` (CLI :`- -new-availability-zones`) sur une liste des zones dans lesquelles vous souhaitez créer les nouveaux nœuds de cache. Pour ElastiCache déterminer la zone de disponibilité de chaque nouveau nœud, ne spécifiez pas de liste.
   - Définissez `ApplyImmediately` (CLI :`- -apply-immediately`) sur true.

   > **Note**
   >
   > Si vous n'utilisez pas la découverte automatique, veillez à mettre à jour votre application cliente avec les nouveaux points de terminaison de nœud de cache.

   Avant de passer à l'étape suivante, assurez-vous que les nœuds Memcached sont entièrement créés et disponibles.

2. Modifiez votre cluster en supprimant les nœuds que vous ne souhaitez plus dans la zone de disponibilité d'origine. Dans votre demande, effectuez les opérations suivantes :

   - Définissez `NumCacheNodes` (CLI :`- -num-cache-nodes`) sur le nombre de nœuds de cache actifs que vous souhaitez après l'application de cette modification.
   - Définissez `CacheNodeIdsToRemove` (CLI :`- -nodes-to-remove`) sur une liste des nœuds de cache que vous souhaitez supprimer du cluster.

     Le nombre de nœuds de cache IDs répertoriés doit être égal au nombre de nœuds actuellement actifs moins la valeur en`NumCacheNodes`.
   - (Facultatif) Définissez `ApplyImmediately` (CLI :`- -apply-immediately`) sur true.

Si vous ne définissez pas `ApplyImmediately` (CLI :`- -apply-immediately`) sur true, les suppressions de nœud auront lieu lors de votre prochaine fenêtre de maintenance.

Localisation de vos nœuds

Amazon ElastiCache prend en charge la localisation de tous les nœuds d'un cluster dans une ou plusieurs zones de disponibilité (AZs). De plus, si vous choisissez de localiser vos nœuds en plusieurs AZs (recommandé), cela vous ElastiCache permet de choisir l'AZ pour chaque nœud ou de les choisir ElastiCache pour vous.

En situant les nœuds dans des emplacements différents AZs, vous éliminez le risque qu'une panne, telle qu'une panne de courant, dans une zone de zone de développement entraîne la défaillance de l'ensemble de votre système. Les tests ont démontré qu'il n'y a pas de différence de latence significative entre le fait de localiser tous les nœuds dans une zone ou de les répartir sur plusieurs AZs.

Vous pouvez spécifier une AZ pour chaque nœud lorsque vous créez un cluster ou en ajoutant des nœuds lorsque vous modifiez un cluster existant. Lorsque vous spécifiez un AZ pour chaque nœud lors de la création d'un cluster, l'AZ doit être disponible dans ce groupe de sous-réseaux. Pour plus d'informations, consultez les ressources suivantes :

- Création d'un cluster pour Memcached
- Création d'un cluster pour Valkey ou Redis OSS
- Modification d'un ElastiCache cluster
- Ajouter des nœuds à un ElastiCache cluster

Régions et terminaux pris en charge

Amazon ElastiCache est disponible dans plusieurs AWS régions. Cela signifie que vous pouvez lancer des ElastiCache clusters 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 particulière pour répondre à certaines exigences légales.

Chaque région est conçue pour être complètement isolée des autres régions . Chaque région dispose de plusieurs zones de disponibilité (AZ). ElastiCache Les caches sans serveur répliquent automatiquement les données sur plusieurs zones de disponibilité (sauf `us-west-1` lorsque les données sont répliquées dans deux zones de disponibilité) pour une haute disponibilité. Lorsque vous concevez votre propre ElastiCache cluster, vous pouvez choisir de lancer vos nœuds différemment AZs pour garantir la tolérance aux pannes. Pour plus d'informations sur les régions et les zones de disponibilité, consultez Choix des régions et des zones de disponibilité pour ElastiCache eau début de cette rubrique.

Régions dans lesquelles ElastiCache est pris en charge

Nom de région/Région	Point de terminaison	Protocole
Région USA Est (Ohio) `us-east-2`	`elasticache.us-east-2.amazonaws.com`	HTTPS
Région USA Est (Virginie du Nord) `us-east-1`	`elasticache.us-east-1.amazonaws.com`	HTTPS
Région US West (N. California) `us-west-1`	`elasticache.us-west-1.amazonaws.com`	HTTPS
Région USA Ouest (Oregon) `us-west-2`	`elasticache.us-west-2.amazonaws.com`	HTTPS
Région Canada (Centre) `ca-central-1`	`elasticache.ca-central-1.amazonaws.com`	HTTPS
Région Canada (Ouest) `ca-west-1`	`elasticache.ca-west-1.amazonaws.com`	HTTPS
Asie-Pacifique (Jakarta) `ap-southeast-3`	`elasticache.ap-southeast-3.amazonaws.com`	HTTPS

Nom de région/Région	Point de terminaison	Protocole
Région Asie-Pacifique (Mumbai) `ap-south-1`	`elasticache.ap-south-1.amazonaws.com`	HTTPS
Région Asie-Pacifique (Hyderabad) `ap-south-2`	`elasticache.ap-south-2.amazonaws.com`	HTTPS
Région Asie-Pacifique (Tokyo) `ap-northeast-1`	`elasticache.ap-northeast-1.amazonaws.com`	HTTPS
Région Asie-Pacifique (Séoul) `ap-northeast-2`	`elasticache.ap-northeast-2.amazonaws.com`	HTTPS
Région Asie-Pacifique (Osaka) `ap-northeast-3`	`elasticache.ap-northeast-3.amazonaws.com`	HTTPS
Région Asie-Pacifique (Singapour) `ap-southeast-1`	`elasticache.ap-southeast-1.amazonaws.com`	HTTPS
Région Asie-Pacifique (Sydney) `ap-southeast-2`	`elasticache.ap-southeast-2.amazonaws.com`	HTTPS

Nom de région/Région	Point de terminaison	Protocole
Région Europe (Francfort) `eu-central-1`	`elasticache.eu-central-1.amazonaws.com`	HTTPS
Région Europe (Zurich) `eu-central-2`	`elasticache.eu-central-2.amazonaws.com`	HTTPS
Région Europe (Stockholm) `eu-north-1`	`elasticache.eu-north-1.amazonaws.com`	HTTPS
Middle East (Bahrain) Region `me-south-1`	`elasticache.me-south-1.amazonaws.com`	HTTPS
Région du Moyen-Orient (EAU) `me-central-1`	`elasticache.me-central-1.amazonaws.com`	HTTPS
Région Europe (Irlande) `eu-west-1`	`elasticache.eu-west-1.amazonaws.com`	HTTPS
Région Europe (Londres) `eu-west-2`	`elasticache.eu-west-2.amazonaws.com`	HTTPS

Nom de région/Région	Point de terminaison	Protocole
Région EU (Paris) `eu-west-3`	`elasticache.eu-west-3.amazonaws.com`	HTTPS
Europe (Milan) Region `eu-south-1`	`elasticache.eu-south-1.amazonaws.com`	HTTPS
Région Europe (Espagne) `eu-south-2`	`elasticache.eu-south-2.amazonaws.com`	HTTPS
Région Amérique du Sud (São Paulo) `sa-east-1`	`elasticache.sa-east-1.amazonaws.com`	HTTPS
Région Chine (Beijing) `cn-north-1`	`elasticache.cn-north-1.amazonaws.com.cn`	HTTPS
Région Chine (Ningxia) `cn-northwest-1`	`elasticache.cn-northwest-1.amazonaws.com.cn`	HTTPS
Région Asie-Pacifique (Hong Kong) `ap-east-1`	`elasticache.ap-east-1.amazonaws.com`	HTTPS

Nom de région/Région	Point de terminaison	Protocole
Région Afrique (Le Cap) `af-south-1`	`elasticache.af-south-1.amazonaws.com`	HTTPS
Région Israël (Tel Aviv) `il-central-1`	`elasticache.il-central-1.amazonaws.com`	HTTPS
AWS GovCloud (US-Ouest) `us-gov-west-1`	`elasticache.us-gov-west-1.amazonaws.com`	HTTPS
AWS GovCloud (USA Est) `us-gov-east-1`	`elasticache.us-gov-east-1.amazonaws.com`	HTTPS
Pour plus d'informations sur l'utilisation du AWS GovCloud (États-Unis) avec ElastiCache, voir Services dans la région AWS GovCloud (États-Unis) : ElastiCache.		

Certaines régions prennent en charge un sous-ensemble de types de nœuds. Pour un tableau des types de nœuds pris en charge par AWS région, consultezTypes de nœuds pris en charge par région AWS.

La plupart des régions prennent en charge l'établissement d'une connexion privée entre votre VPC et les points de terminaison d' ElastiCache API, en créant un point de terminaison VPC d'interface via. AWS PrivateLink Pour de plus amples informations, veuillez consulter ElastiCache Points de terminaison VPC de l'API et de l'interface ()AWS PrivateLink.

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

Utilisation de zones locales avec ElastiCache

Une zone locale est une extension d'une AWS région géographiquement proche de vos utilisateurs. Vous pouvez étendre n'importe quel cloud privé virtuel (VPC) d'une AWS région parent à une zone locale en créant un nouveau sous-réseau et en l'affectant à la zone locale. Lorsque vous créez un sous-réseau dans une zone locale, votre VPC est étendu à cette zone locale. Le sous-réseau de la zone locale fonctionne de la même manière que les autres sous-réseaux de votre VPC.

En utilisant les Zones Locales, vous pouvez placer des ressources telles qu'un ElastiCache cluster à plusieurs endroits proches de vos utilisateurs.

Lorsque vous créez un ElastiCache cluster, vous pouvez choisir un sous-réseau dans une zone locale. Les zones locales ont leurs propres connexions à Internet et prennent en charge AWS Direct Connect. Ainsi, les ressources créées dans une zone locale peuvent servir les utilisateurs locaux avec des communications à très faible latence. Pour de plus amples informations, veuillez consulter AWS Local Zones.

Une zone locale est représentée par un code de AWS région suivi d'un identifiant indiquant l'emplacement, par exemple`us-west-2-lax-1a`.

À l'heure actuelle, les Local Zones disponibles sont `us-west-2-lax-1a` et `us-west-2-lax-1b`.

Les restrictions suivantes s'appliquent ElastiCache aux Zones Locales :

- Les entrepôts de données globaux ne sont pas pris en charge.
- La migration en ligne n'est pas prise en charge.
- Les types de nœuds suivants sont pris en charge par Local Zones pour le moment :
  - Génération actuelle :

    Types de nœuds M5 : `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`, `cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

    Types de nœuds R5: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

    Types de nœuds T3 : `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Activation d'une zone locale

1. Activez la zone locale dans la EC2 console Amazon.

   Pour plus d'informations, consultez la section Activation des zones locales dans le guide de EC2 l'utilisateur Amazon.

2. Créez un sous-réseau dans la zone locale.

   Pour de plus amples informations, veuillez consulter Création d'un sous-réseau dans votre VPC dans le Amazon VPC Guide de l'utilisateur.

3. Créez un groupe de ElastiCache sous-réseaux dans la zone locale.

   Lorsque vous créez un groupe de ElastiCache sous-réseaux, choisissez le groupe de zones de disponibilité pour la zone locale.

   Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

4. Créez un cluster ElastiCache pour Memcached qui utilise le ElastiCache sous-réseau de la zone locale.

   Pour de plus amples informations, veuillez consulter Création d'un cluster Memcached (console).

5. Créez un cluster ElastiCache pour Redis OSS qui utilise le ElastiCache sous-réseau de la zone locale. Pour plus d'informations, consultez l'une des rubriques suivantes :

   - Création d'un cluster Valkey (mode cluster désactivé) (console)
   - Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Utiliser Outposts avec ElastiCache

Vous pouvez utiliser AWS Outposts avec. ElastiCache Outposts est un service entièrement géré qui étend l' AWS infrastructure, les services et les outils aux locaux des clients. APIs En fournissant un accès local à l'infrastructure AWS gérée, AWS Outposts permet aux clients de créer et d'exécuter des applications sur site en utilisant les mêmes interfaces de programmation que dans les AWS régions, tout en utilisant les ressources de calcul et de stockage locales pour réduire la latence et les besoins de traitement des données locaux. Un avant-poste est un pool de capacités de AWS calcul et de stockage déployé sur le site d'un client. AWS exploite, surveille et gère cette capacité dans le cadre d'une AWS région. Vous pouvez créer des sous-réseaux sur votre Outpost et les spécifier lorsque vous créez des AWS ressources telles que ElastiCache des clusters.

> **Note**
>
> Dans cette version, les limitations suivantes s'appliquent :
>
> - ElastiCache for Outposts ne prend en charge que les familles de nœuds M5 et R5.
> - Multi-AZ (la réplication entre Outposts n'est pas prise en charge).
> - La migration active n'est pas prise en charge.
> - Les instantanés locaux ne sont pas pris en charge.
> - Les journaux du moteur et les journaux lents ne peuvent pas être activés.
> - ElastiCache sur Outposts ne supporte pas la CoIP.
> - ElastiCache for Outposts n'est pas pris en charge dans les régions suivantes : cn-north-1, cn-northwest-1 et ap-northeast-3.

Utilisation d'Outposts avec la console ElastiCache

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez les caches Valkey, les caches Redis OSS ou les caches Memcached.
3. Si vous avez choisi les caches Valkey, sélectionnez Créer un cache Valkey. Si vous avez choisi les caches Redis OSS, sélectionnez Créer un cache Redis OSS. Si vous avez choisi des caches Memcached, sélectionnez Créer un cache Memcached.
4. Sous Paramètres du cluster, sélectionnez Concevoir votre propre cache et Cache du cluster. Laissez le mode cluster défini sur Désactivé. Créez ensuite un nom et une description facultative pour le cache.
5. Pour l'emplacement, sélectionnez Sur site.
6. Dans la section Sur site, vous verrez le champ Outpost ID. Entrez l'ID pour lequel le cluster sera exécuté.

   Tous les autres paramètres sous Paramètres du cluster peuvent rester par défaut.
7. Dans Connectivité, sélectionnez Créer un nouveau groupe de sous-réseaux et entrez l'ID du VPC. Conservez le reste comme valeur par défaut, puis sélectionnez Suivant.

Configuration des options sur site

Vous pouvez sélectionner un Outpost disponible pour ajouter votre cluster de cache ou, s'il n'y a pas d'Outposts disponibles, en créer un en procédant comme suit :

Sous On-Premises options (Options sur site) :

1. Dans les paramètres Valkey, les paramètres Redis OSS ou les paramètres Memcached, selon votre moteur préféré :

   a. Nom : entrez un nom pour le cluster
   b. Description : entrez une description pour le cluster.
   c. Compatibilité de la version du moteur : la version du moteur est basée sur la région Outpost AWS
   d. Port : pour Valkey ou Redis OSS, acceptez le port par défaut 6379. Pour Memcached, acceptez le port par défaut 11211. Si vous préférez utiliser un autre port, saisissez le numéro de port.
   e. Parameter group (Groupe de paramètres) : utilisez la liste déroulante pour sélectionner un groupe de paramètres par défaut ou un groupe de paramètres personnalisé.
   f. Node Type (Type de nœud) : les instances disponibles sont basées sur la disponibilité d'Outposts. Si vous utilisez Valkey ou Redis OSS, l'assistant de portage pour .NET for Outposts ne prend en charge que les familles de nœuds M5 et R5. Dans la liste déroulante, sélectionnez Outposts, puis sélectionnez un type de nœud disponible que vous souhaitez utiliser pour ce cluster. Sélectionnez ensuite Save (Enregistrer).
   g. Number of Replicas (Nombre de réplicas) : saisissez le nombre de réplicas en lecture que vous souhaitez créer pour ce groupe de réplication. Vous devez avoir au moins un réplica en lecture et pas plus de cinq. La valeur par défaut est 2.

      Les noms générés automatiquement des réplicas en lecture suivent le même modèle que celui du nom du cluster principal, avec un tiret et un numéro séquentiel à trois chiffres ajoutés à la fin, en commençant par `-002`. Par exemple, si votre groupe de réplication est nommé `MyGroup`, alors les noms des secondaires seraient `MyGroup-002`, `MyGroup-003`, `MyGroup-004`, `MyGroup-005`, `MyGroup-006`.
2. Sous Connectivité :

   a. Subnet Group (Groupe de sous-réseau) : dans la liste, sélectionnez Create new (Créer).

      - Name (Nom) : saisissez un nom pour le groupe de sous-réseaux.

- Description : saisissez une description pour le groupe de sous-réseaux.
- VPC ID (ID de VPC) : l'ID de VPC doit correspondre au VPC Outpost. Si vous sélectionnez un VPC qui n'a aucun sous-réseau IDs dans les Outposts, la liste sera vide.
- Availability Zone or Outpost (Zone de disponibilité ou Outpost) : sélectionnez l'Outpost que vous utilisez.
- Subnet ID ID de sous-réseau) : sélectionnez un ID de sous-réseau disponible pour l'Outpost. Si aucun sous-réseau n' IDs est disponible, vous devez les créer. Pour plus d'informations, consultez Créer un sous-réseau.

b. Sélectionnez Créer.

Affichage des détails du cluster Outpost

Sur la page de liste, sélectionnez un cluster appartenant à un AWS avant-poste et notez les points suivants lorsque vous consultez les détails du cluster :

- Zone de disponibilité : Cela représentera l'avant-poste, en utilisant un ARN (Amazon Resource Name) et le numéro de AWS ressource.
- Nom de l'avant-poste : nom de l' AWS avant-poste.

Utilisation d'Outposts avec la CLI AWS

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour contrôler plusieurs AWS services depuis la ligne de commande et les automatiser par le biais de scripts. Vous pouvez utiliser la AWS CLI pour des opérations ad hoc (ponctuelles).

Téléchargement et configuration du AWS CLI

Il AWS CLI fonctionne sous Windows, macOS ou Linux. Suivez la procédure suivante pour la télécharger et la configurer.

Pour télécharger, installer et configurer la CLI

1. Téléchargez la AWS CLI sur la page Web de l'interface de ligne de AWS commande.
2. Suivez les instructions d'installation de la AWS CLI et de configuration de la AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Utilisation de la AWS CLI avec Outposts

Utilisez l'opération de CLI suivante pour créer un cluster de cache qui utilise Outposts :

- create-cache-cluster— À l'aide de cette opération, le `outpost-mode` paramètre accepte une valeur qui indique si les nœuds du cluster de cache sont créés dans un seul Outpost ou dans plusieurs Outposts.

> **Note**
>
> À l'heure actuelle, seul le mode `single-outpost` est pris en charge.

```
aws elasticache create-cache-cluster \
   --cache-cluster-id cache cluster id \
   --outpost-mode single-outpost \
```

Travailler avec ElastiCache

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

Rubriques

- Instantané et restauration
- Versions du moteur et mise à niveau dans ElastiCache
- ElastiCache meilleures pratiques et stratégies de mise en cache
- Gestion de votre cluster conçu par vos soins dans ElastiCache
- Connexion automatique d'une EC2 instance et d'un ElastiCache cache
- Dimensionnement ElastiCache
- Commencer à utiliser les filtres Bloom
- Commencer à utiliser JSON pour Valkey et Redis OSS
- Marquer vos ressources ElastiCache
- Utilisation de l'objectif Amazon ElastiCache Well-Architected
- Étapes de dépannage courantes et meilleures pratiques avec ElastiCache

Instantané et restauration

ElastiCache Les caches Amazon exécutant Valkey, Redis OSS ou Serverless Memcached peuvent sauvegarder leurs données en créant un instantané. Vous pouvez utiliser la sauvegarde pour restaurer un cache ou des données de départ dans un nouveau cache. La sauvegarde se compose des métadonnées du cache et de toutes les données figurant dans le cache. Toutes les sauvegardes sont écrites sur Amazon Simple Storage Service (Amazon S3), qui fournit un stockage durable. À tout moment, vous pouvez restaurer vos données en créant un nouveau cache Valkey, Redis OSS ou Serverless Memcached et en le remplissant avec les données d'une sauvegarde. Avec ElastiCache, vous pouvez gérer les sauvegardes à l'aide de AWS Management Console, the AWS Command Line Interface (AWS CLI) et de l' ElastiCache API.

Si vous prévoyez de supprimer un cache et qu'il est important de préserver les données, vous pouvez prendre une précaution supplémentaire. Pour ce faire, créez une sauvegarde manuelle, vérifiez que son statut est disponible, puis supprimez le cache. En cas d'échec de la sauvegarde, cette étape

garantit que les données du cache seront toujours disponibles. Vous pouvez réessayer de faire une sauvegarde en suivant les bonnes pratiques énoncées précédemment.

Rubriques

- Contraintes inhérentes à la sauvegarde
- Impact sur les performances des sauvegardes de clusters auto-conçus
- Planification des sauvegardes automatiques
- Réalisation de sauvegardes manuelles
- Création d'une sauvegarde finale
- Description des sauvegardes
- Copie de sauvegardes
- Exportation d'une sauvegarde
- Restauration à partir d'une sauvegarde dans un nouveau cache
- Suppression d'une sauvegarde
- Étiquetage des sauvegardes
- Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe

Contraintes inhérentes à la sauvegarde

Les contraintes suivantes doivent être prises en compte lorsque vous planifiez ou procédez à des procédures de sauvegarde :

- La sauvegarde et la restauration ne sont prises en charge que pour les caches exécutés sur Valkey, Redis OSS ou Serverless Memcached.
- Pour les clusters Valkey ou Redis OSS (mode cluster désactivé), la sauvegarde et la restauration ne sont pas prises en charge sur les `cache.t1.micro` nœuds. Tous les autres types de nœuds de cache sont pris en charge.
- Pour les clusters Valkey ou Redis OSS (mode cluster activé), la sauvegarde et la restauration sont prises en charge pour tous les types de nœuds.
- Au cours d'une période continue de 24 heures, vous ne pouvez pas créer plus de 24 sauvegardes manuelles par cache sans serveur. Pour les clusters conçus par Valkey et Redis OSS, vous ne pouvez pas créer plus de 20 sauvegardes manuelles par nœud du cluster.

- Valkey ou Redis OSS (mode cluster activé) ne prend en charge les sauvegardes qu'au niveau du cluster (pour l'API ou la CLI, au niveau du groupe de réplication). Valkey ou Redis OSS (mode cluster activé) ne prennent pas en charge les sauvegardes au niveau de la partition (pour l'API ou la CLI, au niveau du groupe de nœuds).
- Pendant le processus de sauvegarde, vous ne pouvez exécuter aucune autre opération d'API ou de CLI sur le cache sans serveur. Vous pouvez exécuter des opérations d'API ou de CLI sur un cluster conçu par vos soins pendant la sauvegarde.
- Si vous utilisez des caches Valkey ou Redis OSS avec hiérarchisation des données, vous ne pouvez pas exporter de sauvegarde vers Amazon S3.
- Vous pouvez restaurer une sauvegarde d'un cluster à l'aide du type de nœud r6gd uniquement sur des clusters utilisant le type de nœud r6gd.

Impact sur les performances des sauvegardes de clusters auto-conçus

Les sauvegardes sur les caches sans serveur sont transparentes pour l'application et n'ont aucun impact sur les performances. Cependant, lors de la création de sauvegardes pour des clusters auto-conçus, il peut y avoir un certain impact sur les performances en fonction de la mémoire réservée disponible. Les sauvegardes pour les clusters conçus par vos soins ne sont pas disponibles avec ElastiCache Memcached, mais sont disponibles avec ElastiCache Redis OSS.

Vous trouverez ci-après des instructions pour optimiser les performances des sauvegardes pour les clusters auto-conçus.

- Définir le `reserved-memory-percent` paramètre : pour limiter le pagination excessive, nous vous recommandons de définir le reserved-memory-percentparamètre. Ce paramètre empêche Valkey et Redis OSS de consommer toute la mémoire disponible du nœud et peut contribuer à réduire la quantité de pagination. Vous pouvez également observer une optimisation des performances simplement en utilisant un nœud plus grand. Pour plus d'informations sur la mémoire réservée et les reserved-memory-percentparamètres, consultez. Gestion de la mémoire réservée pour Valkey et Redis OSS

- Créez des sauvegardes à partir d'une réplique en lecture : si vous exécutez Valkey ou Redis OSS dans un groupe de nœuds comportant plusieurs nœuds, vous pouvez effectuer une sauvegarde à partir du nœud principal ou de l'une des répliques en lecture. En raison des ressources système nécessaires au cours d'un BGSAVE, nous vous conseillons de créer des sauvegardes à partir de l'un des réplicas en lecture. Pendant la création de la sauvegarde à partir du réplica, le nœud

principal n'est pas concerné par les exigences de ressources BGSAVE. Le nœud principal peut continuer à répondre aux demandes sans ralentir.

Pour ce faire, veuillez consulter Création d'une sauvegarde manuelle (console) et dans le Cluster Name (Nom du cluster) dans la fenêtre Create Backup (Créer une Backup), choisissez un réplica au lieu du nœud primaire par défaut.

Si vous supprimez un groupe de réplication et demandez une sauvegarde finale, la sauvegarde est ElastiCache toujours prise depuis le nœud principal. Cela garantit que vous capturez les toutes dernières données Valkey ou Redis OSS, avant que le groupe de réplication ne soit supprimé.

Planification des sauvegardes automatiques

Vous pouvez activer les sauvegardes automatiques pour n'importe quel cache sans serveur Valkey ou Redis OSS ou pour tout cluster conçu par vos soins. Lorsque les sauvegardes automatiques sont activées, ElastiCache crée une sauvegarde du cache sur une base quotidienne. Il n'y a aucun impact sur le cache et le changement est immédiat. Les sauvegardes automatiques peuvent constituer une protection contre la perte de données. En prévision d'une éventuelle défaillance, vous pouvez créer un cache et restaurer vos données à partir de la sauvegarde la plus récente. Vous obtenez ainsi un cache démarré à chaud, dans lequel vos données ont été préchargées, et qui est prêt à l'emploi. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.

Vous pouvez activer les sauvegardes automatiques pour n'importe quel cache Memcached Serverless. Lorsque les sauvegardes automatiques sont activées, ElastiCache crée une sauvegarde du cache sur une base quotidienne. Il n'y a aucun impact sur le cache et le changement est immédiat. Les sauvegardes automatiques peuvent constituer une protection contre la perte de données. En prévision d'une éventuelle défaillance, vous pouvez créer un cache et restaurer vos données à partir de la sauvegarde la plus récente. Vous obtenez ainsi un cache démarré à chaud, dans lequel vos données ont été préchargées, et qui est prêt à l'emploi. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.

Lorsque vous planifiez des sauvegardes automatiques, vous devez prévoir les paramètres suivants :

- Heure de début de la sauvegarde : heure de la journée ElastiCache à laquelle commence la création d'une sauvegarde. Vous pouvez définir le créneau de sauvegarde qui vous convient le mieux. Si vous ne spécifiez pas de fenêtre de sauvegarde, en ElastiCache assigne une automatiquement.

- Limite de conservation des sauvegardes – Nombre de jours pendant lesquels la sauvegarde sera conservée dans Amazon S3. Par exemple, si vous définissez la limite de conservation à 5, une sauvegarde effectuée aujourd'hui sera conservée pendant 5 jours. Lorsque la limite de conservation expire, la sauvegarde est supprimée automatiquement.

  La limite de conservation maximale des sauvegardes est de 35 jours. Si la limite de conservation des sauvegardes est définie sur 0, les sauvegardes automatiques sont désactivées pour le cache.

Lorsque vous planifiez des sauvegardes automatiques, ElastiCache la création de la sauvegarde commence. Vous pouvez définir le créneau de sauvegarde qui vous convient le mieux. Si vous ne spécifiez pas de fenêtre de sauvegarde, en ElastiCache assigne une automatiquement.

Vous pouvez activer ou désactiver les sauvegardes automatiques lors de la création d'un nouveau cache ou de la mise à jour d'un cache existant, à l'aide de la ElastiCache console, de l' AWS CLI API ou de l' ElastiCache API. Pour Valkey et Redis OSS, cela se fait en cochant la case Activer les sauvegardes automatiques dans la section Paramètres avancés de Valkey ou Paramètres avancés de Redis OSS. Pour Memcached, cela se fait en cochant la case Activer les sauvegardes automatiques dans la section Paramètres avancés de Memcached.

Réalisation de sauvegardes manuelles

Outre les sauvegardes automatiques, vous pouvez créer une sauvegarde manuelle à tout moment. Contrairement aux sauvegardes automatiques, qui sont automatiquement supprimées après une période de conservation spécifiée, les sauvegardes manuelles ne sont pas soumises à une période de conservation et ne sont donc pas automatiquement supprimées. Même si vous supprimez le cache, toutes les sauvegardes manuelles de ce cache sont conservées. Si vous ne souhaitez plus conserver une sauvegarde manuelle, vous devez la supprimer par vous-même.

Les sauvegardes manuelles peuvent non seulement être créées directement, mais également de l'une des façons suivantes :

- Copie de sauvegardes. Peu importe si la sauvegarde source a été créée automatiquement ou manuellement.
- Création d'une sauvegarde finale. Créez une sauvegarde juste avant de supprimer un cluster ou un nœud.

Vous pouvez créer une sauvegarde manuelle d'un cache à l'aide de AWS Management Console, de AWS CLI, ou de l' ElastiCache API.

Vous pouvez générer des sauvegardes manuelles à partir de répliques dont le mode cluster est activé et le mode cluster est désactivé.

Création d'une sauvegarde manuelle (console)

Pour créer une sauvegarde d'un cache (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez les caches Valkey, les caches Redis OSS ou les caches Memcached, selon vos préférences.
3. Cochez la case située à gauche du nom du cache que vous souhaitez sauvegarder.
4. Choisissez Backup.
5. Dans la boîte de dialogue Créer une sauvegarde, tapez le nom de votre sauvegarde dans la boîte Nom de la sauvegarde. Il est préférable que le nom comporte une indication du cluster qui a été sauvegardé, ainsi que la date et l'heure auxquelles la sauvegarde a été effectuée.

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. Choisissez Create Backup.

   L'état du cluster devient snapshotting.

Création d'une sauvegarde manuelle (AWS CLI)

Sauvegarde manuelle d'un cache sans serveur à l'aide du AWS CLI

Pour créer une sauvegarde manuelle d'un cache à l'aide de AWS CLI, utilisez l'`create-serverless-snapshot` AWS CLI opération avec les paramètres suivants :

- `--serverless-cache-name` : nom du cache sans serveur que vous sauvegardez.
- `--serverless-cache-snapshot-name` – Nom de l'instantané à créer.

Pour Linux, macOS ou Unix :

- ```
  aws elasticache create-serverless-snapshot \
                        --serverless-cache-name CacheName \
                        --serverless-cache-snapshot-name bkup-20231127
  ```

Pour Windows :

- ```
  aws elasticache create-serverless-snapshot ^
      --serverless-cache-name CacheName ^
      --serverless-cache-snapshot-name bkup-20231127
  ```

Sauvegarde manuelle d'un cluster conçu par vos soins à l'aide du AWS CLI

Pour créer une sauvegarde manuelle d'un cluster conçu par vos soins à l'aide de AWS CLI, utilisez l'`create-snapshot` AWS CLI opération avec les paramètres suivants :

- `--cache-cluster-id`
  - Si le cluster que vous sauvegardez ne possède pas de nœuds de réplication, `--cache-cluster-id` c'est le nom du cluster que vous sauvegardez, par exemple*`mycluster`*.
  - Si le cluster que vous sauvegardez a un ou plusieurs nœuds de réplica, `--cache-cluster-id` est le nom du nœud du cluster que vous voulez utiliser pour la sauvegarde. Par exemple, le nom peut être*`mycluster-002`*.

  Utilisez ce paramètre uniquement lors de la sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster désactivé).

- `--replication-group-id`— Nom du cluster Valkey ou Redis OSS (mode cluster activé) (CLI/API : un groupe de réplication) à utiliser comme source pour la sauvegarde. Utilisez ce paramètre lors de la sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster activé).

- `--snapshot-name` – Nom de l'instantané à créer.

  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.

Exemple 1 : sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster désactivé) qui ne possède aucun nœud de réplication

L' AWS CLI opération suivante crée la sauvegarde `bkup-20150515` à partir du cluster Valkey ou Redis OSS (mode cluster désactivé) `myNonClusteredRedis` qui ne possède aucune réplique en lecture.

Pour Linux, macOS ou Unix :

```
aws elasticache create-snapshot \
    --cache-cluster-id myNonClusteredRedis \
    --snapshot-name bkup-20150515
```

Pour Windows :

```
aws elasticache create-snapshot ^
    --cache-cluster-id myNonClusteredRedis ^
    --snapshot-name bkup-20150515
```

Exemple 2 : sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds répliques

L' AWS CLI opération suivante crée la sauvegarde `bkup-20150515` à partir du cluster Valkey ou Redis OSS (mode cluster désactivé). `myNonClusteredRedis` Cette sauvegarde a un ou plusieurs réplicas en lecture.

Pour Linux, macOS ou Unix :

```
aws elasticache create-snapshot \
    --cache-cluster-id myNonClusteredRedis-001 \
    --snapshot-name bkup-20150515
```

Pour Windows :

```
aws elasticache create-snapshot ^
    --cache-cluster-id myNonClusteredRedis-001 ^
    --snapshot-name bkup-20150515
```

Exemple de sortie : sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds répliqués

Le résultat de cette opération ressemble à ceci.

```
{
    "Snapshot": {
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis6.x",
        "VpcId": "vpc-91280df6",
        "CacheClusterId": "myNonClusteredRedis-001",
        "SnapshotRetentionLimit": 0,
        "NumCacheNodes": 1,
        "SnapshotName": "bkup-20150515",
        "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",
        "AutoMinorVersionUpgrade": true,
```

```
        "PreferredAvailabilityZone": "us-east-1c",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "08:30-09:30",
        "EngineVersion": "6.0",
        "NodeSnapshots": [
            {
                "CacheSize": "",
                "CacheNodeId": "0001",
                "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"
            }
        ],
        "CacheSubnetGroupName": "default",
        "Port": 6379,
        "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
        "CacheNodeType": "cache.m3.2xlarge",
        "DataTiering": "disabled"
    }
}
```

Exemple 3 : sauvegarde d'un cluster pour Valkey ou Redis OSS (mode cluster activé)

L' AWS CLI opération suivante crée la sauvegarde `bkup-20150515` à partir du cluster Valkey ou Redis OSS (mode cluster activé). `myClusteredRedis` Notez l'utilisation de `--replication-group-id` au lieu de `--cache-cluster-id` pour identifier la source. Notez également que la sauvegarde ElastiCache est effectuée à l'aide du nœud de réplication lorsqu'il est présent, et qu'il utilise par défaut le nœud principal si un nœud de réplication n'est pas disponible.

Pour Linux, macOS ou Unix :

```
aws elasticache create-snapshot \
    --replication-group-id myClusteredRedis \
    --snapshot-name bkup-20150515
```

Pour Windows :

```
aws elasticache create-snapshot ^
    --replication-group-id myClusteredRedis ^
    --snapshot-name bkup-20150515
```

Exemple de sortie : sauvegarde d'un cluster Valkey ou Redis OSS (mode cluster activé)

Le résultat de cette opération ressemble à ceci.

```
{
    "Snapshot": {
        "Engine": "redis",
        "CacheParameterGroupName": "default.redis6.x.cluster.on",
        "VpcId": "vpc-91280df6",
        "NodeSnapshots": [
            {
                "CacheSize": "",
                "NodeGroupId": "0001"
            },
            {
                "CacheSize": "",
                "NodeGroupId": "0002"
            }
        ],
        "NumNodeGroups": 2,
        "SnapshotName": "bkup-20150515",
        "ReplicationGroupId": "myClusteredRedis",
        "AutoMinorVersionUpgrade": true,
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotStatus": "creating",
        "SnapshotSource": "manual",
        "SnapshotWindow": "10:00-11:00",
        "EngineVersion": "6.0",
        "CacheSubnetGroupName": "default",
        "ReplicationGroupDescription": "2 shards 2 nodes each",
        "Port": 6379,
        "PreferredMaintenanceWindow": "sat:03:30-sat:04:30",
        "CacheNodeType": "cache.r3.large",
        "DataTiering": "disabled"
    }
}
```

Rubriques en relation

Pour de plus amples informations, veuillez consulter create-bucket dans la Référence des commandes de la AWS CLI .

Création d'une sauvegarde à l'aide de AWS CloudFormation

Vous pouvez l'utiliser AWS CloudFormation pour créer une sauvegarde de votre cache ElastiCache Redis OSS ou Valkey à l'aide des propriétés `AWS::ElastiCache::ServerlessCache` or`AWS::ElastiCache::ReplicationGroup`.

Utilisation de la **`AWS::ElastiCache::ServerlessCache`** ressource

Utilisez-le pour créer une sauvegarde à l'aide de la AWS::ElastiCache::ServerlessCache ressource :

```
Resources:
                    iotCatalog:
                        Type: AWS::ElastiCache::ServerlessCache
                            Properties:
                            ...
                            ServerlessCacheName: "your-cache-name"
                            Engine: "redis"
                            CacheUsageLimits
```

Utilisation de la AWS::ElastiCache::ReplicationGroup ressource

Utilisez la `AWS::ElastiCache::ReplicationGroup` ressource :

```
Resources:
                    iotCatalog:
                        Type: AWS::ElastiCache::ReplicationGroup
                            Properties:
                            ...
                            ReplicationGroupDescription: "Description of your
 replication group"
                            Engine: "redis"
                            CacheNodeType
                            NumCacheClusters
                            AutomaticFailoverEnabled
                            AtRestEncryptionEnabled
```

Création d'une sauvegarde finale

Vous pouvez créer une sauvegarde finale à l'aide de la ElastiCache console, de AWS CLI, ou de l' ElastiCache API.

Création d'une sauvegarde finale (console)

Vous pouvez créer une sauvegarde finale lorsque vous supprimez un cache sans serveur Valkey, Memcached ou Redis OSS, ou un cluster personnalisé Valkey ou Redis OSS, à l'aide de la console. ElastiCache

Pour créer une sauvegarde finale lors de la suppression d'un cache, dans la boîte de dialogue de suppression, sélectionnez Oui sous Créer une sauvegarde et nommez la sauvegarde.

Rubriques en relation

- En utilisant le AWS Management Console
- Suppression d'un groupe de réplication (console)

Création d'une sauvegarde finale (AWS CLI)

Vous pouvez créer une sauvegarde finale lorsque vous supprimez un cache à l'aide du AWS CLI.

Rubriques

- Lors de la suppression d'un cache Valkey, d'un cache sans serveur Memcached ou d'un cache Redis OSS
- Lors de la suppression d'un cluster Valkey ou Redis OSS sans répliques en lecture
- Lors de la suppression d'un cluster Valkey ou Redis OSS avec des répliques de lecture

Lors de la suppression d'un cache Valkey, d'un cache sans serveur Memcached ou d'un cache Redis OSS

Pour créer une sauvegarde finale, utilisez l'`delete-serverless-cache` AWS CLI opération avec les paramètres suivants.

- `--serverless-cache-name` : nom du cache en cours de suppression.
- `--final-snapshot-name` – Nom de la sauvegarde.

Le code suivant permet de créer la sauvegarde finale `bkup-20231127-final` lors de la suppression du cache `myserverlesscache`.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-serverless-cache \
        --serverless-cache-name myserverlesscache \
        --final-snapshot-name bkup-20231127-final
```

Pour Windows :

```
aws elasticache delete-serverless-cache ^
        --serverless-cache-name myserverlesscache ^
        --final-snapshot-name bkup-20231127-final
```

Pour plus d'informations, consultez delete-serverless-cache dans la Référence des commandes de l'AWS CLI .

Lors de la suppression d'un cluster Valkey ou Redis OSS sans répliques en lecture

Pour créer une sauvegarde finale pour un cluster conçu par vos soins sans répliques en lecture, utilisez l'`delete-cache-cluster` AWS CLI opération avec les paramètres suivants.

- `--cache-cluster-id` – Nom du cluster en cours de suppression.
- `--final-snapshot-identifier` – Nom de la sauvegarde.

Le code suivant crée la sauvegarde finale `bkup-20150515-final` lors de la suppression du cluster `myRedisCluster`.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-cluster \
        --cache-cluster-id myRedisCluster \
        --final-snapshot-identifier bkup-20150515-final
```

Pour Windows :

```
aws elasticache delete-cache-cluster ^
        --cache-cluster-id myRedisCluster ^
```

```
        --final-snapshot-identifier bkup-20150515-final
```

Pour plus d'informations, consultez delete-cache-cluster dans la Référence des commandes de l'AWS CLI .

Lors de la suppression d'un cluster Valkey ou Redis OSS avec des répliques de lecture

Pour créer une sauvegarde finale lors de la suppression d'un groupe de réplication, utilisez l'`delete-replication-group` AWS CLI opération, avec les paramètres suivants :

- `--replication-group-id` – Nom du groupe de réplication en cours de suppression.
- `--final-snapshot-identifier` – Nom de la sauvegarde finale.

Le code suivant déclenche la sauvegarde finale `bkup-20150515-final` lors de la suppression du groupe de réplication `myReplGroup`.

Pour Linux, macOS ou Unix :

```
aws elasticache delete-replication-group \
        --replication-group-id myReplGroup \
        --final-snapshot-identifier bkup-20150515-final
```

Pour Windows :

```
aws elasticache delete-replication-group ^
        --replication-group-id myReplGroup ^
        --final-snapshot-identifier bkup-20150515-final
```

Pour plus d'informations, consultez delete-replication-group dans la Référence des commandes de l'AWS CLI .

Description des sauvegardes

Les procédures suivantes montrent comment afficher la liste de vos sauvegardes. Si vous le souhaitez, vous pouvez également afficher les détails d'une sauvegarde particulière.

Description des sauvegardes (console)

Pour afficher les sauvegardes à l'aide du AWS Management Console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes.
3. Pour consulter les détails d'une sauvegarde particulière, choisissez la case située à gauche du nom de la sauvegarde.

Description de sauvegardes sans serveur (AWS CLI)

Pour afficher une liste des sauvegardes sans serveur et, le cas échéant, les détails relatifs à une sauvegarde spécifique, utilisez l'opération d'interface de ligne de commande `describe-serverless-cache-snapshots`.

Exemples

L'opération suivante utilise le paramètre `--max-records` pour afficher jusqu'à 20 sauvegardes associées à votre compte. Si le paramètre `--max-records` n'est pas spécifié, la liste répertorie jusqu'à 50 sauvegardes.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

L'opération suivante utilise le paramètre `--serverless-cache-name` pour répertorier uniquement les sauvegardes associées au cache `my-cache`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

L'opération suivante utilise le paramètre `--serverless-cache-snapshot-name` pour afficher les détails de la sauvegarde `my-backup`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-
name my-backup
```

Pour plus d'informations, consultez describe-serverless-cache-snapshotsle manuel de référence des AWS CLI commandes.

Description de sauvegardes de cluster auto-conçu (AWS CLI)

Pour afficher une liste des sauvegardes de cluster auto-conçu et, le cas échéant, les détails relatifs à une sauvegarde spécifique, utilisez l'opération d'interface de ligne de commande `describe-snapshots`.

Exemples

L'opération suivante utilise le paramètre `--max-records` pour afficher jusqu'à 20 sauvegardes associées à votre compte. Si le paramètre `--max-records` n'est pas spécifié, la liste répertorie jusqu'à 50 sauvegardes.

```
aws elasticache describe-snapshots --max-records 20
```

L'opération suivante utilise le paramètre `--cache-cluster-id` pour afficher uniquement les sauvegardes associées au cluster `my-cluster`.

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

L'opération suivante utilise le paramètre `--snapshot-name` pour afficher les détails de la sauvegarde `my-backup`.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Pour plus d'informations, consultez la section describe-snapshots dans le AWS CLI manuel de référence des commandes.

Copie de sauvegardes

Vous pouvez créer une copie de n'importe quelle sauvegarde, qu'elle ait été créée automatiquement ou manuellement. Vous pouvez également exporter votre sauvegarde afin d'y accéder de l'extérieur ElastiCache. Pour obtenir de l'aide lors de l'exportation de votre sauvegarde, consultez Exportation d'une sauvegarde.

Les étapes suivantes expliquent comment copier une sauvegarde.

Copie de sauvegardes (console)

Pour copier une sauvegarde (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes pour afficher la liste de vos sauvegardes.
3. Dans la liste des sauvegardes, cochez la case située à gauche du nom de la sauvegarde à copier.
4. Choisissez Actions, puis Copier.
5. Saisissez un nom pour votre nouvelle sauvegarde dans la zone New Backup name.
6. Choisissez Copier.

Copie d'une sauvegarde sans serveur (AWS CLI)

Pour copier une sauvegarde d'un cache sans serveur, utilisez l'opération `copy-serverless-cache-snapshot`.

Paramètres

- `--source-serverless-cache-snapshot-name` – Nom de la sauvegarde à copier.
- `--target-serverless-cache-snapshot-name` – Nom de la copie de la sauvegarde.

L'exemple suivant fait une copie d'une sauvegarde automatique.

Pour Linux, macOS ou Unix :

```
aws elasticache copy-serverless-cache-snapshot \
```

```
    --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \
    --target-serverless-cache-snapshot-name my-backup-copy
```

Pour Windows :

```
aws elasticache copy-serverless-cache-snapshot ^
    --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^
    --target-serverless-cache-snapshot-name my-backup-copy
```

Pour de plus amples informations, veuillez consulter `copy-serverless-cache-snapshot` dans le AWS CLI.

Copie d'une sauvegarde de cluster auto-conçu (AWS CLI)

Pour copier une sauvegarde d'un cluster auto-conçu, utilisez l'opération `copy-snapshot`.

Paramètres

- `--source-snapshot-name` – Nom de la sauvegarde à copier.
- `--target-snapshot-name` – Nom de la copie de la sauvegarde.
- `--target-bucket` – Réservé à l'exportation d'une sauvegarde. N'utilisez pas ce paramètre lorsque vous copiez une sauvegarde. Pour de plus amples informations, veuillez consulter Exportation d'une sauvegarde.

L'exemple suivant fait une copie d'une sauvegarde automatique.

Pour Linux, macOS ou Unix :

```
aws elasticache copy-snapshot  \
    --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \
    --target-snapshot-name amzn-s3-demo-bucket
```

Pour Windows :

```
aws elasticache copy-snapshot ^
    --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^
    --target-snapshot-name amzn-s3-demo-bucket
```

Pour de plus amples informations, veuillez consulter `copy-snapshot` dans le AWS CLI.

Exportation d'une sauvegarde

Amazon ElastiCache prend en charge l'exportation de votre sauvegarde ElastiCache pour Redis OSS vers un bucket Amazon Simple Storage Service (Amazon S3), ce qui vous permet d'y accéder depuis l'extérieur. ElastiCache Vous pouvez exporter une sauvegarde à l'aide de la ElastiCache console AWS CLI, de l'API ou de l' ElastiCache API.

L'exportation d'une sauvegarde 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 cache au lieu d'attendre que le nouveau cluster soit rempli en cours d'utilisation. Pour de plus amples informations sur l'amorçage d'un nouveau cluster, veuillez consulter Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe. Une autre raison pour laquelle vous souhaiterez peut-être exporter les données de votre cache est d'utiliser le fichier .rdb pour le traitement hors ligne.

> **Important**
>
> - La ElastiCache sauvegarde et le compartiment Amazon S3 dans lequel vous souhaitez la copier doivent se trouver dans la même AWS région.
>
> Bien que les sauvegardes copiées sur un compartiment Amazon S3 soient chiffrées, nous vous conseillons vivement de ne pas autoriser l'accès au compartiment Amazon S3 dans lequel vous souhaitez stocker vos sauvegardes.
> - L'exportation d'une sauvegarde vers Amazon S3 n'est pas prise en charge pour les clusters utilisant la hiérarchisation des données. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.
> - L'exportation d'une sauvegarde est disponible pour les clusters auto-conçus par Valkey et Redis OSS, Serverless Valkey et Redis OSS, et Serverless Memcached. L'exportation d'une sauvegarde n'est pas disponible pour les clusters Memcached conçus par nos soins.

Avant de pouvoir exporter une sauvegarde vers un compartiment Amazon S3, vous devez disposer d'un compartiment Amazon S3 dans la même AWS région que la sauvegarde. Accordez l' ElastiCache accès au bucket. Les deux premières étapes vous indiquent comment procéder.

Créer un compartiment Amazon S3

Les étapes suivantes utilisent la console Amazon S3 pour créer un compartiment Amazon S3 dans lequel vous exportez et stockez votre ElastiCache sauvegarde.

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.

      Le nom de votre compartiment Amazon S3 doit être conforme au DNS. Dans le cas contraire, ElastiCache vous ne pourrez 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).

   b. Dans la liste des régions, choisissez une AWS région pour votre compartiment Amazon S3. Cette AWS région doit être la même AWS que celle de la ElastiCache sauvegarde 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.

Accordez l' ElastiCache accès à votre compartiment Amazon S3

ElastiCache Pour pouvoir copier un instantané dans un compartiment Amazon S3, vous devez mettre à jour votre politique de compartiment IAM afin d'autoriser l' ElastiCache accès au compartiment.

> **Warning**
>
> Bien que les sauvegardes copiées dans un compartiment Amazon S3 soient chiffrées, vos données sont accessibles à toute personne ayant accès à votre compartiment Amazon S3. De ce fait, nous vous recommandons vivement de configurer les politiques IAM afin d'empêcher tout accès non autorisé à ce compartiment Amazon S3. Pour plus d'informations, consultez Gestion des accès dans le Guide de l'utilisateur Amazon S3.

Pour créer les autorisations appropriées sur un compartiment Amazon S3, procédez comme suit.

Pour accorder ElastiCache 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 la sauvegarde. Il doit s'agir du compartiment S3 que vous avez créé dans Créer un compartiment Amazon S3.
3. Choisissez l'onglet Permissions (Autorisations) et sous Permissions (Autorisations), choisissez Access control list (ACL) (Liste de contrôle d'accès (ACL)), puis choisissez Edit (Modifier).
4. Ajoutez l'identifiant canonique du bénéficiaire `540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353` avec les options suivantes :
   - Objets : Liste, Écriture
   - ACL de compartiment : Lecture, Écriture

   > **Note**
   >
   > - Pour la GovCloud région PDT, l'identifiant canonique est. `40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6`

- Pour la GovCloud région OSU, l'identifiant canonique est. `c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c`

5. Choisissez Enregistrer.

Exporter une ElastiCache sauvegarde

Vous avez maintenant créé votre compartiment S3 et vous avez accordé ElastiCache les autorisations nécessaires pour y accéder. Ensuite, vous pouvez utiliser la ElastiCache console, la AWS CLI ou l' ElastiCache API pour y exporter votre instantané.

La politique mise à jour devrait ressembler à l'exemple suivant.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399484",
            "Effect": "Allow",

            "Principal": {
                "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        }
    ]
}
```

Pour les régions optionnelles, voici un exemple de ce à quoi pourrait ressembler la politique IAM mise à jour pour le compartiment S3. (Cet exemple utilise la région Asie-Pacifique (Hong Kong).)

JSON

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticache.amazonaws.com"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        },
        {
            "Sid": "Stmt15399484",
            "Effect": "Allow",

            "Principal": {
                "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::hkg-elasticache-backup",
                "arn:aws:s3:::hkg-elasticache-backup/*"
            ]
        }
    ]
}
```

Exportation d'une ElastiCache sauvegarde (console)

Les étapes suivantes utilisent la ElastiCache console pour exporter une sauvegarde vers un compartiment Amazon S3 afin que vous puissiez y accéder depuis l'extérieur ElastiCache. Le compartiment Amazon S3 doit se trouver dans la même AWS région que la ElastiCache sauvegarde.

Pour exporter une ElastiCache sauvegarde vers un compartiment Amazon S3

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes pour afficher la liste de vos sauvegardes.
3. Dans la liste des sauvegardes, choisissez la case située à gauche du nom de la sauvegarde à exporter.
4. Choisissez Copier.
5. Dans Create Copy of the Backup? (Créer une copie de la sauvegarde ?), procédez comme suit :

   a. Saisissez un nom pour votre nouvelle sauvegarde dans la zone New Backup name.

      Le nom doit comprendre entre 1 et 1 000 caractères et pouvoir être encodé en UTF-8.

      ElastiCache ajoute un identifiant d'instance et `.rdb` à la valeur que vous entrez ici. Par exemple, si vous entrez `my-exported-backup`, ElastiCache crée `my-exported-backup-0001.rdb`.

   b. Dans la liste Emplacement S3 cible, choisissez le nom du compartiment Amazon S3 dans lequel vous souhaitez copier votre sauvegarde (le compartiment que vous avez créé dans Créer un compartiment Amazon S3).

      L'emplacement S3 cible doit être un compartiment Amazon S3 situé dans la AWS région de la sauvegarde avec les autorisations suivantes pour que le processus d'exportation réussisse.

      - Accès à l'objet – Lecture et Écriture.
      - Accès aux autorisations – Lecture.

      Pour de plus amples informations, veuillez consulter Accordez l' ElastiCache accès à votre compartiment Amazon S3.

   c. Choisissez Copier.

> **Note**
>
> Si votre compartiment S3 ne dispose pas des autorisations nécessaires ElastiCache pour y exporter une sauvegarde, vous recevez l'un des messages d'erreur suivants. Revenez à Accordez l' ElastiCache accès à votre compartiment Amazon S3 pour ajouter les autorisations spécifiées et essayer à nouveau d'exporter votre sauvegarde.
>
> - ElastiCache n'a pas obtenu les autorisations READ %s sur le compartiment S3.
>
> Solution : ajoutez des autorisations de lecture sur le compartiment.
> - ElastiCache n'a pas obtenu les autorisations WRITE %s sur le compartiment S3.
>
> Solution : ajoutez des autorisations d'écriture sur le compartiment.
> - ElastiCache 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 sauvegarde vers une autre AWS région, utilisez Amazon S3 pour la copier. Pour de plus amples informations, veuillez consulter Copie d'objets dans le Guide de l'utilisateur Amazon Simple Storage Service.

Exportation d'une sauvegarde ElastiCache sans serveur ()AWS CLI

Exportation d'une sauvegarde d'un cache sans serveur

Exportez la sauvegarde dans un compartiment Amazon S3 à l'aide de l'opération `export-serverless-cache-snapshot` de la CLI avec les paramètres suivants :

Paramètres

- `--serverless-cache-snapshot-name` – Nom de la sauvegarde à copier.
- `--s3-bucket-name` – Nom du compartiment Amazon S3 dans lequel vous voulez exporter la sauvegarde. Une copie de la sauvegarde est effectuée dans le compartiment spécifié.

  `--s3-bucket-name`Il doit s'agir d'un compartiment Amazon S3 dans la AWS région de la sauvegarde avec les autorisations suivantes pour que le processus d'exportation réussisse.
  - Accès à l'objet – Lecture et Écriture.
  - Accès aux autorisations – Lecture.

L'opération suivante copie une sauvegarde dans le compartiment my-s3-bucket.

Pour Linux, macOS ou Unix :

```
aws elasticache export-serverless-cache-snapshot \
    --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \
    --s3-bucket-name my-s3-bucket
```

Pour Windows :

```
aws elasticache export-serverless-cache-snapshot ^
    --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^
    --s3-bucket-name my-s3-bucket
```

Exportation d'une sauvegarde de ElastiCache cluster conçue par vos soins ()AWS CLI

Exportation d'une sauvegarde d'un cluster auto-conçu

Exportez la sauvegarde dans un compartiment Amazon S3 à l'aide de l'opération `copy-snapshot` de la CLI avec les paramètres suivants :

Paramètres

- `--source-snapshot-name` – Nom de la sauvegarde à copier.
- `--target-snapshot-name` – Nom de la copie de la sauvegarde.

  Le nom doit comprendre entre 1 et 1 000 caractères et pouvoir être encodé en UTF-8.

  ElastiCache ajoute un identifiant d'instance et `.rdb` à la valeur que vous entrez ici. Par exemple, si vous entrez `my-exported-backup`, ElastiCache crée `my-exported-backup-0001.rdb`.
- `--target-bucket` – Nom du compartiment Amazon S3 dans lequel vous voulez exporter la sauvegarde. Une copie de la sauvegarde est effectuée dans le compartiment spécifié.

  `--target-bucket`Il doit s'agir d'un compartiment Amazon S3 dans la AWS région de la sauvegarde avec les autorisations suivantes pour que le processus d'exportation réussisse.
  - Accès à l'objet – Lecture et Écriture.
  - Accès aux autorisations – Lecture.

  Pour de plus amples informations, veuillez consulter [Accordez l' ElastiCache accès à votre compartiment Amazon S3].

L'opération suivante copie une sauvegarde dans le compartiment my-s3-bucket.

Pour Linux, macOS ou Unix :

```
aws elasticache copy-snapshot \
    --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \
    --target-snapshot-name my-exported-backup \
    --target-bucket my-s3-bucket
```

Pour Windows :

```
aws elasticache copy-snapshot ^
    --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^
    --target-snapshot-name my-exported-backup ^
    --target-bucket my-s3-bucket
```

Restauration à partir d'une sauvegarde dans un nouveau cache

Vous pouvez restaurer une sauvegarde existante depuis Valkey dans un nouveau cache Valkey ou un cluster conçu par vos soins, et restaurer une sauvegarde Redis OSS existante dans un nouveau cache Redis OSS ou un cluster conçu par vos soins. Vous pouvez également restaurer une sauvegarde de cache sans serveur Memcached existante dans un nouveau cache Memcached sans serveur.

Restauration d'une sauvegarde dans un cache sans serveur (console)

> **Note**
>
> ElastiCache Serverless prend en charge les fichiers RDB compatibles avec Valkey 7.2 et versions ultérieures, ainsi que les versions Redis OSS comprises entre la version 5.0 et la dernière version disponible.

Pour restaurer une sauvegarde sur un cache sans serveur (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes.
3. Dans la liste des sauvegardes, cochez la case située à gauche du nom de la sauvegarde à restaurer.
4. Choisissez Actions, puis Restaurer.
5. Entrez un nom et une description facultative pour le nouveau cache sans serveur.
6. Cliquez sur Créer pour créer votre cache et importer les données de votre sauvegarde.

Restauration d'une sauvegarde dans un cluster auto-conçu (console)

Pour restaurer une sauvegarde dans un cluster auto-conçu (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation de gauche, choisissez Sauvegardes.
3. Dans la liste des sauvegardes, cochez la case située à gauche du nom de la sauvegarde à restaurer.

4. Choisissez Actions, puis Restaurer.
5. Choisissez Concevez votre propre cache et personnalisez les paramètres du cluster, tels que le type de nœud, les tailles, le nombre de partitions, les réplicas, le placement des zones de disponibilité et les paramètres de sécurité.
6. Choisissez Créer pour créer votre cache auto-conçu et importer les données de votre sauvegarde.

Restauration d'une sauvegarde dans un cache sans serveur (AWS CLI)

> **Note**
>
> ElastiCache Serverless prend en charge les fichiers RDB compatibles avec Valkey 7.2 et versions ultérieures, ainsi que les versions Redis OSS comprises entre la version 5.0 et la dernière version disponible.

Pour restaurer une sauvegarde sur un nouveau cache sans serveur (AWS CLI)

L' AWS CLI exemple suivant crée un nouveau cache en utilisant les données d'une sauvegarde `create-serverless-cache` et en importe les données.

Pour Linux, macOS ou Unix :

```
aws elasticache create-serverless-cache \

    --serverless-cache-name CacheName \
    --engine redis
    --snapshot-arns-to-restore Snapshot-ARN
```

Pour Windows :

```
aws elasticache create-serverless-cache ^

    --serverless-cache-name CacheName ^
    --engine redis ^
    --snapshot-arns-to-restore Snapshot-ARN
```

Restauration d'une sauvegarde dans un cluster auto-conçu (AWS CLI)

Pour restaurer une sauvegarde sur un cluster auto-conçu (AWS CLI)

Vous pouvez restaurer une sauvegarde Valkey ou Redis OSS (mode cluster désactivé) de deux manières.

- ```
  aws elasticache create-serverless-cache \
      --serverless-cache-name CacheName \
      --engine redis
      --snapshot-arns-to-restore Snapshot-ARN
  ```
- Pour Windows :

  ```
  aws elasticache create-serverless-cache ^
      --serverless-cache-name CacheName ^
      --engine redis ^
      --snapshot-arns-to-restore Snapshot-ARN
  ```

Restauration d'une sauvegarde dans un cluster auto-conçu (AWS CLI)

Pour restaurer une sauvegarde sur un cluster auto-conçu (AWS CLI)

Vous pouvez restaurer une sauvegarde de cache sans serveur Valkey ou Redis OSS, et vous pouvez également restaurer un cluster conçu par Valkey ou Redis OSS.

Vous pouvez restaurer une sauvegarde de cache sans serveur Valkey ou Redis OSS de deux manières.

- Vous pouvez effectuer une restauration vers un cluster Valkey ou Redis OSS à nœud unique (mode cluster désactivé) à l'aide de cette opération. AWS CLI create-cache-cluster
- Vous pouvez effectuer une restauration sur un cluster Valkey ou Redis OSS avec des répliques de lecture (un groupe de réplication). Pour ce faire, vous pouvez utiliser Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé) avec l'opération. AWS CLI create-replication-group Dans ce cas, vous amorcez la restauration avec un fichier .rdb Valkey ou Redis OSS. Pour plus d'informations sur l'alimentation d'un nouveau cluster auto-conçu, consultez Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.

Vous pouvez restaurer une sauvegarde Valkey ou Redis OSS (mode cluster désactivé) de deux manières.
  ```

- Vous pouvez effectuer une restauration vers un cluster Valkey ou Redis OSS à nœud unique (mode cluster désactivé) à l'aide de cette opération. AWS CLI create-cache-cluster
- Vous pouvez effectuer une restauration sur un cluster Valkey ou Redis OSS avec des répliques de lecture (un groupe de réplication). Pour ce faire, vous pouvez utiliser Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé) avec l'opération. AWS CLI create-replication-group Dans ce cas, vous amorcez la restauration avec un fichier .rdb Valkey ou Redis OSS. Pour plus d'informations sur l'alimentation d'un nouveau cluster auto-conçu, consultez Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.

Lorsque vous utilisez l'opération `create-cache-cluster` ou `create-replication-group`, veillez à inclure le paramètre `--snapshot-name` ou `--snapshot-arn` pour alimenter le nouveau cluster ou groupe de réplication avec les données de la sauvegarde.

Suppression d'une sauvegarde

Une sauvegarde automatique est supprimée automatiquement à l'expiration de sa limite de conservation. Si vous supprimez un cluster, toutes ses sauvegardes automatiques sont également supprimées. Si vous supprimez un groupe de réplication, toutes les sauvegardes automatiques des clusters de ce groupe sont également supprimées.

ElastiCache fournit une opération d'API de suppression qui vous permet de supprimer une sauvegarde à tout moment, que la sauvegarde ait été créée automatiquement ou manuellement. Puisque les sauvegardes manuelles n'ont pas de limite de conversation, une suppression manuelle est le seul moyen de les supprimer.

Vous pouvez supprimer une sauvegarde à l'aide de la ElastiCache console AWS CLI, de l'API ou de l' ElastiCache API.

Suppression d'une sauvegarde (console)

La procédure suivante permet de supprimer une sauvegarde à l'aide de la ElastiCache console.

Pour supprimer une sauvegarde

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Sauvegardes.

La fenêtre Backups affiche une liste de vos sauvegardes.

3. Cochez la case située à gauche du nom de la sauvegarde à supprimer.
4. Sélectionnez Delete (Supprimer).
5. Si vous souhaitez supprimer cette sauvegarde, sélectionnez Supprimer sur l'écran de confirmation Supprimer une sauvegarde. L'état passe à deleting.

Suppression d'une sauvegarde sans serveur (AWS CLI)

Utilisez l' AWS CLI opération delete-snapshot avec le paramètre suivant pour supprimer une sauvegarde sans serveur.

- `--serverless-cache-snapshot-name` – Nom de la sauvegarde à supprimer.

Le code suivant supprime la sauvegarde `myBackup`.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-
name myBackup
```

Pour plus d'informations, consultez delete-serverless-cache-snapshot dans la Référence des commandes de l'AWS CLI .

Suppression d'une sauvegarde de cluster auto-conçu (AWS CLI)

Utilisez l' AWS CLI opération delete-snapshot avec le paramètre suivant pour supprimer une sauvegarde de cluster conçue par vos soins.

- `--snapshot-name` – Nom de la sauvegarde à supprimer.

Le code suivant supprime la sauvegarde `myBackup`.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Pour plus d'informations, veuillez consulter delete-snapshot dans la Référence des commandes AWS CLI .

Étiquetage des sauvegardes

Vous pouvez attribuer vos propres métadonnées à chaque sauvegarde sous la forme de balises. Les balises vous permettent de classer vos ressources de différentes manières, par exemple, par objectif, par propriétaire ou par 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 de plus amples informations, veuillez consulter 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 ElastiCache console AWS CLI, de l' ElastiCache API ou de l'API, vous pouvez ajouter, répertorier, modifier, supprimer ou copier des balises de répartition des coûts sur vos sauvegardes. Pour de plus amples informations, veuillez consulter Surveillance des coûts avec des balises de répartition des coûts.

Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe

Lorsque vous créez un nouveau cluster personnalisé Valkey ou Redis OSS, vous pouvez l'amorcer avec les données d'un fichier de sauvegarde .rdb Valkey ou Redis OSS. L'ensemencement du cluster est utile si vous gérez actuellement une instance Valkey ou Redis OSS en dehors de ElastiCache et si vous souhaitez remplir votre nouveau ElastiCache cluster conçu par Redis OSS avec vos données Valkey ou Redis OSS existantes.

Pour démarrer un nouveau cluster personnalisé Valkey ou Redis OSS à partir d'une sauvegarde Valkey ou Redis OSS créée sur Amazon, consultez. ElastiCache Restauration à partir d'une sauvegarde dans un nouveau cache

Lorsque vous utilisez un fichier .rdb Valkey ou Redis OSS pour démarrer un nouveau cluster conçu par vos soins, vous pouvez effectuer les opérations suivantes :

- Passez d'un cluster non partitionné à un cluster personnalisé Valkey ou Redis OSS (mode cluster activé) exécutant Redis OSS version 3.2.4.
- Spécifier un nombre de partitions (appelées groupes de nœuds dans l'API et l'interface de ligne de commande) dans le nouveau cluster auto-conçu. Ce nombre peut être différent du nombre de partitions dans le cluster auto-conçu utilisé pour créer le fichier de sauvegarde.
- Spécifier un type de nœud différent pour le nouveau cluster auto-conçu, plus grand ou plus petit que celui utilisé dans le cluster qui a réalisé la sauvegarde. 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 Valkey ou Redis OSS. Pour de plus amples informations, veuillez consulter S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS.
- Répartissez vos clés dans les emplacements du nouveau cluster Valkey ou Redis OSS (mode cluster activé) différemment que dans le cluster utilisé pour créer le fichier de sauvegarde.

> **Note**
>
> Vous ne pouvez pas démarrer un cluster Valkey ou Redis OSS (mode cluster désactivé) à partir d'un fichier .rdb créé à partir d'un cluster Valkey ou Redis OSS (mode cluster activé).

Important

- Vous devez vous assurer que vos données de sauvegarde Valkey ou Redis OSS ne dépassent pas les ressources du nœud. Par exemple, vous ne pouvez pas télécharger un fichier .rdb contenant 5 Go de données Valkey ou Redis OSS vers un nœud cache.m3.medium doté de 2,9 Go de mémoire.

  Si la sauvegarde est trop volumineuse, le cluster obtenu aura le statut `restore-failed`. Si cela se produit, vous devez supprimer le cluster et recommencer.

  Pour une liste complète des types de nœuds et de leurs spécifications, consultez Paramètres spécifiques au type de nœud Redis OSS la section Caractéristiques et détails des ElastiCache produits Amazon.
- Vous pouvez chiffrer un fichier .rdb Valkey ou Redis OSS 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.

Vous trouverez ci-dessous des rubriques qui vous guident dans la migration de votre cluster depuis l'extérieur ElastiCache pour Valkey ou Redis OSS vers ElastiCache Redis OSS.

Migration vers Redis ElastiCache OSS

- Étape 1 : créer une sauvegarde Valkey ou Redis OSS
- Étape 2 : Créer un compartiment et un dossier Amazon S3
- Étape 3 : Charger votre sauvegarde sur Amazon S3
- Étape 4 : Accorder l'accès en ElastiCache lecture au fichier .rdb

Migration depuis des services externes vers ElastiCache Redis OSS.

- Étape 1 : créer une sauvegarde Valkey ou Redis OSS
- Étape 2 : Créer un compartiment et un dossier Amazon S3
- Étape 3 : Charger votre sauvegarde sur Amazon S3
- Étape 4 : Accorder l'accès en ElastiCache lecture au fichier .rdb

Étape 1 : créer une sauvegarde Valkey ou Redis OSS

Pour créer la sauvegarde Valkey ou Redis OSS afin de démarrer votre ElastiCache instance Redis OSS

1. Connectez-vous à votre instance Valkey ou Redis OSS existante.
2. Exécutez l'`BGSAVE`une ou `SAVE` l'autre opération pour créer une sauvegarde. 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 Valkey.

   `SAVE` est synchrone et bloque les autres processus jusqu'à la fin. Pour plus d'informations, consultez SAVE sur le site Web de Valkey.

Pour plus d'informations sur la création d'une sauvegarde, consultez Persistence sur le site Web de Valkey.

Étape 2 : Créer un compartiment et un dossier Amazon S3

Une fois que vous avez créé le fichier de sauvegarde, vous devez le télécharger dans un dossier du 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 : Charger votre sauvegarde 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. Sinon, ElastiCache vous ne pourrez 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).

Vous devez créer votre compartiment Amazon S3 dans la même AWS région que votre nouveau ElastiCache cluster Redis OSS. Cette approche garantit la vitesse de transfert de données la plus élevée lors de la ElastiCache lecture de 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 Valkey ou Redis OSS.

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 : Charger votre sauvegarde sur Amazon S3

Maintenant, téléchargez le fichier .rdb que vous avez créé dans Étape 1 : créer une sauvegarde Valkey ou Redis OSS. 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, veuillez consulter Ajouter un objet à un compartiment. 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. Choisissez 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 bons fichiers figurent dans la boîte de dialogue Charger, 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 d'accès pour alimenter le nouveau cluster avec les données de cette sauvegarde.

Pour de plus amples informations, veuillez consulter Limites et restrictions applicables aux compartiments dans le Guide de l'utilisateur Amazon Simple Storage Service.

Étape 4 : Accorder l'accès en ElastiCache lecture au fichier .rdb

Accordez maintenant l'accès en ElastiCache lecture à votre fichier de sauvegarde .rdb. Vous accordez ElastiCache l'accès à votre fichier de sauvegarde d'une manière différente selon que votre compartiment se trouve dans une AWS région par défaut ou dans une AWS région optionnelle.

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, telles que l'Asie-Pacifique (Hong Kong) et Moyen-Orient (Bahreïn), sont désactivées par défaut. Vous devez activer ces régions, ou vous y inscrire, avant de pouvoir les utiliser, comme décrit dans la rubrique Gestion des régions AWS dans le document Références générales AWS.

Choisissez votre approche en fonction de votre AWS région :

- Pour une région par défaut, utilisez la procédure de Accorder l'accès en ElastiCache lecture au fichier .rdb dans une région par défaut.
- Pour une région pour laquelle une inscription est nécessaire, utilisez la procédure de Accorder l'accès en ElastiCache lecture au fichier .rdb dans une région optionnelle.

Accorder l'accès en ElastiCache lecture au fichier .rdb dans une région par défaut

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, telles que l'Asie-Pacifique (Hong Kong) et Moyen-Orient (Bahreïn), sont désactivées par défaut. Vous devez activer ces régions, ou vous y inscrire, avant de pouvoir les utiliser, comme décrit dans la rubrique Gestion des régions AWS dans le document Références générales AWS.

Pour accorder un accès en ElastiCache lecture au fichier de sauvegarde dans une AWS région activée par défaut

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 sauvegarde .rdb. Le nom du fichier sélectionné apparaît au-dessus des onglets en haut de la page.
5. Choisissez Autorisations.
6. Si aws-scs-s3-readonly ou l'un des modèles canoniques de la liste suivante n'est pas répertorié IDs en tant qu'utilisateur, procédez comme suit :

    a. Sous Accès pour les autres AWS comptes, choisissez Ajouter un bénéficiaire.

    b. Dans le champ, ajoutez l'identifiant canonique de la AWS région comme indiqué ci-dessous :

        - AWS GovCloud Région (ouest des États-Unis) :

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

> **Important**
>
> La sauvegarde doit se trouver dans un compartiment S3 AWS GovCloud (US) pour que vous puissiez la télécharger sur un cluster Valkey ou Redis OSS. AWS GovCloud (US)

- AWS Régions activées par défaut :

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

c. Définissez les autorisations sur le compartiment en choisissant Oui pour :

- Objet en liste/écriture
- Autorisations ACL de l'objet en lecture/écriture

d. Choisissez Enregistrer.

7. Choisissez Présentation, puis Télécharger.

Accorder l'accès en ElastiCache lecture au fichier .rdb dans une région optionnelle

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, telles que l'Asie-Pacifique (Hong Kong) et Moyen-Orient (Bahreïn), sont désactivées par défaut. Vous devez activer ces régions, ou vous y inscrire, avant de pouvoir les utiliser, comme décrit dans la rubrique Gestion des régions AWS dans le document Références générales AWS.

Accordez maintenant l'accès en ElastiCache lecture à votre fichier de sauvegarde .rdb.

Pour accorder un accès en ElastiCache lecture au fichier de sauvegarde

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 sauvegarde .rdb. Le nom du fichier sélectionné apparaît au-dessus des onglets en haut de la page.
5. Sélectionnez l'onglet Autorisations.

6. Sous Permissions (Autorisations), choisissez Bucket policy (Politique de compartiment), puis Edit (Modifier).
7. Mettez à jour la politique pour accorder les autorisations ElastiCache requises pour effectuer des opérations :
   - Ajoutez `[ "Service" : "region-full-name.elasticache-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.

   JSON

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/backup1.rdb",
                "arn:aws:s3:::amzn-s3-demo-bucket/backup2.rdb"
            ]
        }
    ]
```

```
}
```

8. Sélectionnez Enregistrer les modifications.

Ensemencez le ElastiCache cluster avec les données du fichier .rdb

Vous êtes maintenant prêt à créer un ElastiCache cluster et à l'ensemencer avec les données du fichier .rdb. Pour créer le cluster, suivez les instructions de Création d'un cluster pour Valkey ou Redis OSS ou Création d'un groupe de réplication Valkey ou Redis OSS à partir de zéro. Assurez-vous de choisir Valkey ou Redis OSS comme moteur de cluster.

La méthode que vous utilisez pour savoir ElastiCache où trouver la sauvegarde que vous avez téléchargée sur Amazon S3 dépend de la méthode que vous avez utilisée pour créer le cluster :

Ensemencez le ElastiCache cluster ou le groupe de réplication Redis OSS avec les données du fichier .rdb

- Utilisation de la ElastiCache console

  Lorsque vous sélectionnez Cluster settings (Paramètres du cluster), choisissez Restore from backups (Restaurer à partir de sauvegardes) comme méthode de création du cluster, puis choisissez Other backups (Autres sauvegardes) comme Source dans la section Backup source (Source de sauvegarde). 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-cache-cluster` ou `create-replication-group`, 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 résoudre les fichiers de sauvegarde stockés dans Amazon S3.

- Utilisation de l' ElastiCache API

  Si vous utilisez l'opération `CreateCacheCluster` ou l'`CreateReplicationGroup` ElastiCache API, 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 résoudre les fichiers de sauvegarde stockés dans Amazon S3.

> **Important**
>
> Lors de l'amorçage d'un cluster Valkey ou Redis OSS (mode cluster activé), vous devez configurer chaque groupe de nœuds (partition) dans le nouveau cluster ou groupe de réplication. Pour ce faire, utilisez le paramètre `--node-group-configuration` (API : `NodeGroupConfiguration`). Pour plus d'informations, consultez les ressources suivantes :
>
> - CLI : create-replication-groupdans la AWS CLI référence
> - API : CreateReplicationGroupdans la référence de ElastiCache l'API

Au cours du processus de création de votre cluster, les données de votre sauvegarde Valkey ou Redis OSS sont écrites dans le cluster. Vous pouvez suivre la progression en consultant les messages des ElastiCache événements. Pour ce faire, consultez la ElastiCache console et choisissez Cache Events. Vous pouvez également utiliser l'interface de ligne de AWS ElastiCache commande ou l' ElastiCache API pour obtenir des messages d'événements. Pour de plus amples informations, veuillez consulter Affichage des ElastiCache événements.

Versions du moteur et mise à niveau dans ElastiCache

Cette section décrit les moteurs Valkey, Memcached et Redis OSS pris en charge et explique comment procéder à la mise à niveau. Notez que toutes les fonctionnalités disponibles avec Redis OSS 7.2 sont disponibles par défaut dans Valkey 7.2 et versions ultérieures. Vous pouvez également passer de certains moteurs Redis OSS existants ElastiCache à un moteur Valkey.

Mise à niveau des versions du moteur, y compris les mises à niveau entre

Valkey et Redis OSS

Avec Valkey et Redis OSS, vous initiez les mises à niveau de version de votre cluster ou groupe de réplication en le modifiant à l'aide de la ElastiCache console, de l'API ou de l' AWS CLI ElastiCache API et en spécifiant une version du moteur plus récente.

Vous pouvez également effectuer une mise à niveau croisée de Redis OSS vers Valkey. Pour plus d'informations sur les mises à niveau croisées, consultezComment passer de Redis OSS à Valkey.

Rubriques

- Comment passer de Redis OSS à Valkey
- Résolution des mises à niveau bloquées du moteur Valkey ou Redis OSS

Comment modifier des clusters et des groupes de réplication	
Caches	Groupes de réplication
À l'aide du ElastiCache AWS Management Console	À l'aide du AWS Management Console
En utilisant le AWS CLI avec ElastiCache	À l'aide du AWS CLI
Utilisation de l' ElastiCache API	Utilisation de l' ElastiCache API

Memcached

Avec Memcached, pour démarrer les mises à niveau de version de votre cluster, vous devez le modifier et spécifier une version du moteur plus récente. Vous pouvez le faire en utilisant la ElastiCache console, le AWS CLI, ou l' ElastiCache API :

- Pour utiliser le AWS Management Console, voir —À l'aide du ElastiCache AWS Management Console.
- Pour utiliser le AWS CLI, voirEn utilisant le AWS CLI avec ElastiCache.
- Pour utiliser l' ElastiCache API, consultezUtilisation de l' ElastiCache API.

Comment passer de Redis OSS à Valkey

Valkey est conçu pour remplacer directement Redis OSS 7. Vous pouvez effectuer une mise à niveau de Redis OSS vers Valkey à l'aide de la console, de l'API ou de la CLI, en spécifiant le nouveau moteur et la version principale du moteur. L'adresse IP du point de terminaison et tous les autres aspects de l'application ne seront pas modifiés par la mise à niveau. Lors de la mise à niveau depuis Redis OSS 5.0.6 ou version ultérieure, vous ne subirez aucun temps d'arrêt.

> **Note**
>
> AWS Exigences relatives à la version de la CLI pour les mises à niveau de Redis OSS vers Valkey :
>
> - Pour la AWS CLI v1 : version minimale requise 1.35.2 (version actuelle : 1.40.22)
> - Pour la AWS CLI v2 : version minimale requise 2.18.2 (version actuelle : 2.27.22)

> **Note**
>
> - Lors de la mise à niveau à partir de versions antérieures de Redis OSS à 5.0.6, vous pouvez rencontrer un temps de basculement de 30 à 60 secondes pendant la propagation du DNS.
> - Pour mettre à niveau un cluster à nœud unique Redis OSS (mode cluster désactivé) existant vers le moteur Valkey, procédez d'abord comme suit :. Création d'un groupe de réplication à l'aide d'un cluster existant Une fois que le cluster à nœud unique Redis OSS (mode cluster désactivé) a été ajouté à un groupe de réplication, vous pouvez effectuer une mise à niveau entre moteurs vers Valkey.

Mise à niveau d'un groupe de réplication de Redis OSS vers Valkey

Si vous avez un groupe de réplication Redis OSS existant qui utilise le groupe de paramètres de cache par défaut, vous pouvez passer à Valkey en spécifiant le nouveau moteur et la nouvelle version du moteur avec modify-replication-group l'API.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --engine valkey \
   --engine-version 8.0
```

Pour Windows :

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --engine valkey ^
   --engine-version 8.0
```

Si un groupe de paramètres de cache personnalisé est appliqué au groupe de réplication Redis OSS existant que vous souhaitez mettre à niveau, vous devrez également transmettre un groupe de paramètres de cache Valkey personnalisé dans la demande. Le groupe de paramètres personnalisés Valkey en entrée doit avoir les mêmes valeurs de paramètres statiques Redis OSS que le groupe de paramètres personnalisés Redis OSS existant.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --engine valkey \
   --engine-version 8.0 \
   --cache-parameter-group-name myParamGroup
```

Pour Windows :

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --engine valkey ^
   --engine-version 8.0 ^
   --cache-parameter-group-name myParamGroup
```

Mise à niveau d'un cache sans serveur Redis OSS vers Valkey avec la CLI

Pour Linux, macOS ou Unix :

```
aws elasticache modify-serverless-cache \
   --serverless-cache-name myCluster \
   --engine valkey \
   --major-engine-version 8
```

Pour Windows :

```
aws elasticache modify-serverless-cache ^
   --serverless-cache-name myCluster ^
   --engine valkey ^
   --major-engine-version 8
```

Mise à niveau de Redis OSS vers Valkey avec la console

Mise à niveau de Redis OSS 5 vers Valkey

1. Sélectionnez le cache Redis OSS à mettre à niveau.
2. Une fenêtre de mise à niveau vers Valkey devrait apparaître. Cliquez sur le bouton Upgrade to Valkey.
3. Accédez aux paramètres du cache, puis sélectionnez Version du moteur. La version la plus récente de Valkey est recommandée.
4. Si ce cache est sans serveur, vous devez mettre à jour le groupe de paramètres. Accédez à la zone Groupes de paramètres des paramètres du cache, sélectionnez un groupe de paramètres approprié tel que default.valkey8.
5. Sélectionnez Mettre à niveau.

Ce cache sera désormais répertorié dans la zone Valkey de la console.

> **Note**
>
> La mise à niveau directe de Redis OSS 4 ou d'une version antérieure vers Valkey peut entraîner un temps de basculement plus long de 30 à 60 secondes pendant la propagation du DNS.

Résolution des mises à niveau bloquées du moteur Valkey ou Redis OSS

Comme indiqué dans le tableau suivant, votre opération de mise à niveau du moteur Valkey ou 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 l'échelle ascendante
Mise à niveau du moteur	Mise à niveau du moteur
Augmentation et mise à niveau du moteur	Mise à niveau du moteur
	Mise à l'échelle ascendante

Pour résoudre une mise à niveau bloquée du moteur Valkey ou Redis OSS

- Effectuez l'une des actions suivantes :
  - Planifiez votre opération de mise à niveau du moteur Redis OSS ou Valkey pour la prochaine fenêtre de maintenance en décochant la case Appliquer immédiatement.

    Avec la CLI, utilisez `--no-apply-immediately`. Avec l'API, utilisez `ApplyImmediately=false`.
  - Attendez votre prochaine fenêtre de maintenance (ou après) pour effectuer l'opération de mise à niveau du moteur Redis OSS.
  - Ajoutez l'opération de mise à l'échelle Redis OSS à cette modification de cluster en cochant la case Appliquer immédiatement.

    Avec la CLI, utilisez `--apply-immediately`. Avec l'API, utilisez `ApplyImmediately=true`.

    Cette approche annule efficacement la mise à niveau du moteur lors de la prochaine fenêtre de maintenance en l'effectuant immédiatement.

ElastiCache Support étendu

Grâce ElastiCache au Support étendu, vous pouvez continuer à exécuter votre cache sur une version majeure du moteur après la date de fin du support standard moyennant des frais supplémentaires. Si vous ne procédez pas à la mise à niveau après la date de fin du support standard, des frais vous seront facturés.

Le support étendu fournit les mises à jour et le support technique suivants :

- Mises à jour de sécurité critiques et élevées CVEs pour votre cache et votre moteur de cache
- Corrections de bogues et correctifs pour les problèmes critiques
- Possibilité d'ouvrir des dossiers de support et de recevoir de l'aide au dépannage dans le cadre du contrat ElastiCache de niveau de service standard

Cette offre payante vous donne plus de temps pour passer à une version majeure du moteur compatible.

Par exemple, la date de ElastiCache fin du support standard pour Redis OSS 4.0.10 est le 31 janvier 2026. Si vous n'êtes pas prêt à passer manuellement à Valkey ou à Redis OSS 6 ou version ultérieure à cette date, vos caches ElastiCache seront automatiquement inscrits au Support étendu et vous pourrez continuer à exécuter Redis OSS 4.0.10. À compter du premier jour du mois suivant la fin du support standard, le 1er février 2026, le support étendu vous ElastiCache sera automatiquement facturé.

Le support étendu est disponible jusqu'à 3 ans après la date de fin du support standard pour une version majeure du moteur. Pour les versions 4 et 5 d'Elasticache pour Redis OSS, ce sera le 31 janvier 2029. Après cette date, tous les caches qui exécutent encore les versions 4 et 5 de Redis OSS seront automatiquement mis à niveau vers la dernière version de Valkey.

Une fois la période de support d'un moteur terminée, les caches qui continuent à exécuter cette ancienne version passeront automatiquement au support étendu. Vous serez averti avant la date de début des tarifs du Support étendu afin que vous puissiez mettre à niveau votre instance à la place. Vous pouvez également vous désinscrire explicitement à tout moment en effectuant une mise à niveau vers les versions prises en charge.

Pour plus d'informations sur les dates de fin du support standard et les dates de fin du support étendu, consultez ElastiCache versions pour le calendrier de fin de vie de Redis OSS Valkey, Memcached ou Redis OSS.

Rubriques

- ElastiCache Frais de Support étendu
- Versions avec Support ElastiCache étendu
- ElastiCache et responsabilités du client avec ElastiCache Extended Support

ElastiCache Frais de Support étendu

Vous devrez payer des frais pour tous les moteurs inscrits au Support ElastiCache étendu à compter du lendemain de la fin du support standard. Pour connaître la date de ElastiCache fin du support standard, consultezVersions avec Support ElastiCache étendu.

Les frais supplémentaires pour le Support ElastiCache étendu s'arrêtent automatiquement lorsque vous effectuez l'une des actions suivantes :

- Passez à une version du moteur couverte par le support standard.
- Supprimez le cache qui exécute une version majeure après la date de ElastiCache fin de support standard.

Les frais reprendront si la version de votre moteur cible bénéficie du Support étendu à l'avenir.

Par exemple, supposons que la ElastiCache version 4 de Redis OSS entre dans le support étendu le 1er février 2026 et que vous mettez à niveau vos caches de la v4 vers la v6 le 1er janvier 2027. Vous ne serez facturé que pour 11 mois de Support étendu, sur la ElastiCache version 4 de Redis OSS. Si vous continuez à exécuter ElastiCache la version 6 pour Redis OSS après la date de fin du support standard du 31 janvier 2027, ces caches seront à nouveau soumis à des frais de support étendu à compter du 1er février 2027.

Vous pouvez éviter de payer des frais de ElastiCacheExtended support en ElastiCache empêchant la création ou la restauration d'un cache après la date de ElastiCache fin du support standard.

Pour plus d'informations, consultez les ElastiCache tarifs Amazon.

Versions avec Support ElastiCache étendu

Les versions 4 et 5 du logiciel libre Redis (OSS) ont atteint la fin de vie de leur communauté en 2020 et 2022, respectivement. Cela signifie qu'aucune autre mise à jour, correction de bogue ou correctif de sécurité n'est publiée par la communauté. Le support standard pour les versions 4 et 5 de ElastiCache Redis OSS ElastiCache prendra fin le 31 janvier 2026. Si vous continuez à utiliser

des versions non prises en charge de Redis OSS, vos données risquent d'être vulnérables à des vulnérabilités et à des expositions courantes connues (). CVEs

À compter du 1er février 2026, ElastiCache les caches toujours en cours d'exécution sur les versions 4 et 5 de Redis OSS seront automatiquement inscrits au Support étendu, afin de garantir une disponibilité et une sécurité continues. Bien que le support étendu offre de la flexibilité, nous vous recommandons de considérer la fin du support standard comme une étape de planification de vos charges de travail de production. Nous vous encourageons vivement à mettre à niveau vos caches Redis OSS v4 et v5 vers ElastiCache Valkey ou Redis OSS v6 ou version ultérieure, avant la fin du support standard.

Le tableau suivant récapitule la date de ElastiCache fin du support standard d'Amazon et les dates du support étendu.

Support étendu et calendrier de fin de vie

Version principale du moteur	Support standard terminé	Début du Support étendu Y1 Premium	Début du Support étendu Y2 Premium	Début du Support étendu Y3 Premium	Fin du support étendu et de la version EOL
Redis OSS v4	31/01/2026	01/02/2026	01/02/2027	01/02/2028	31/01/2029
Redis OSS v5	31/01/2026	01/02/2026	01/02/2027	01/02/2028	31/01/2029
Redis OSS 6	31/01/2027	01/02/2027	01/02/2028	01/02/2029	31/01/2030

Support étendu ne sera proposé que pour la dernière version de correctif prise en charge de chaque version majeure de Redis OSS. Lorsque le support étendu débutera le 1er février 2026, si vos clusters Redis OSS v4 et v5 ne disposent pas déjà des dernières versions de correctif, ils seront automatiquement mis à niveau vers la version 4.0.10 pour Redis OSS v4 et vers la version 5.0.6 pour Redis OSS v5, avant d'être inscrits au support étendu. Cela garantit que vous recevrez des mises à jour de sécurité et des corrections de bogues par le biais du Support étendu. Vous n'avez aucune action à effectuer pour passer à ces dernières versions de correctifs dans le cadre de la transition vers le Support étendu.

ElastiCache et responsabilités du client avec ElastiCache Extended Support

Vous trouverez ci-dessous les responsabilités d'Amazon ElastiCache et vos responsabilités en matière de Support ElastiCache étendu.

ElastiCache Responsabilités d'Amazon

Après la ElastiCache fin de la date de support standard, Amazon ElastiCache fournira des correctifs, des corrections de bogues et des mises à niveau pour les moteurs inscrits au support ElastiCache étendu. Cela se produira pendant 3 ans au maximum, ou jusqu'à ce que vous arrêtiez d'utiliser les moteurs dans le cadre du Support étendu, selon la première éventualité.

Vos responsabilités

Vous êtes responsable de l'application des correctifs, des corrections de bogues et des mises à niveau proposés pour les caches dans le cadre du Support ElastiCache étendu. Amazon ElastiCache se réserve le droit de modifier, de remplacer ou de retirer ces correctifs, corrections de bogues et mises à niveau à tout moment. Si un correctif est nécessaire pour résoudre des problèmes de sécurité ou de stabilité critiques, Amazon ElastiCache se réserve le droit de mettre à jour vos caches avec le correctif ou d'exiger que vous installiez le correctif.

Vous êtes également responsable de la mise à niveau de votre moteur vers une version plus récente avant la ElastiCache fin de la date de Support étendu. La date de ElastiCache fin du support étendu est généralement 3 ans après la date de ElastiCache fin du support standard.

Si vous ne mettez pas à niveau votre moteur, après la ElastiCache fin de la date de support étendu, Amazon ElastiCache tentera de le mettre à niveau vers une version plus récente prise en charge dans le cadre du support ElastiCache standard. Si la mise à niveau échoue, Amazon ElastiCache se réserve le droit de supprimer le cache qui exécute le moteur après la date de ElastiCache fin du support standard. Cependant, avant de le faire, Amazon ElastiCache préservera vos données provenant de ce moteur.

Gestion des versions pour ElastiCache

Gérez la manière dont vous souhaitez mettre à jour vos ElastiCache caches et vos clusters conçus par vous-même et mis à jour pour les moteurs Valkey, Memcached et Redis OSS.

Gestion des versions pour ElastiCache Serverless Cache

Gérez si et quand le cache ElastiCache sans serveur est mis à niveau et effectuez les mises à niveau de version selon vos propres conditions et délais.

ElastiCache Serverless applique automatiquement la dernière version mineure et corrective du logiciel à votre cache, sans aucun impact ni interruption de service pour votre application. Aucune action de votre part n'est nécessaire.

Lorsqu'une nouvelle version majeure est disponible, ElastiCache Serverless vous envoie une notification dans la console et un événement dans EventBridge. Vous pouvez choisir de mettre à niveau votre cache vers la dernière version majeure en modifiant votre cache à l'aide de la console, de l'interface de ligne de commande ou de l'API et en sélectionnant la dernière version du moteur. À l'instar des mises à niveau mineures et des correctifs, les mises à niveau des versions majeures sont effectuées sans interruption de service de votre application.

Gestion des versions pour les clusters conçus par vos soins ElastiCache

Lorsque vous travaillez avec des ElastiCache clusters conçus par vous-même, vous pouvez contrôler le moment où le logiciel qui alimente votre cluster de cache est mis à niveau vers les nouvelles versions prises en charge par ElastiCache. Vous pouvez contrôler à quel moment mettre à niveau votre cache vers les dernières versions majeures, mineures et patchs disponibles. Vous lancez les mises à niveau de version du moteur dans votre cluster ou groupe de réplication en le modifiant et en spécifiant une nouvelle version de moteur.

Vous pouvez contrôler si et quand le logiciel conforme au protocole qui alimente votre cluster de cache est mis à niveau vers de nouvelles versions prises en charge par. ElastiCache 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.

Comme les mises à niveau de version peuvent présenter un risque en termes de compatibilité, elles ne se produisent pas automatiquement. Vous devez les initier.

Clusters Valkey et Redis OSS

> **Note**
>
> - Si un cluster Valkey ou Redis OSS est répliqué dans une ou plusieurs régions, la version du moteur est mise à niveau pour les régions secondaires, puis pour la région principale.
> - ElastiCache pour Redis OSS, les versions sont identifiées par une version sémantique comprenant un composant majeur et un composant mineur. Par exemple, dans Redis OSS 6.2, la version principale est 6 et la version mineure 2. Lorsque vous utilisez des clusters

conçus par vos soins, ElastiCache Redis OSS expose également le composant correctif, par exemple Redis OSS 6.2.1, et la version du correctif est 1.

Les versions majeures concernent les modifications incompatibles avec l'API et les versions mineures sont destinées aux nouvelles fonctionnalités ajoutées de manière rétrocompatible. Les versions de correctif sont destinées à corriger des bogues rétrocompatibles et à apporter des modifications non fonctionnelles.

Avec Valkey et Redis OSS, vous initiez les mises à niveau de version du moteur vers votre cluster ou groupe de réplication en le modifiant et en spécifiant une nouvelle version du moteur. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.

Memcached

Avec Memcached, pour passer à une version plus récente, vous devez modifier votre cluster de cache et spécifier la nouvelle version du moteur que vous souhaitez utiliser. La mise à niveau vers une version Memcached plus récente est un processus destructeur – vous perdez vos données et repartez avec un cache passif. Pour de plus amples informations, veuillez consulter Modification d'un ElastiCache cluster.

Vous devez être conscient des exigences suivantes quand vous effectuez une mise à niveau à partir d'une ancienne version de Memcached vers la version 1.4.33 ou une version ultérieure. `CreateCacheCluster` et `ModifyCacheCluster` échouent dans les conditions suivantes :

- Si `slab_chunk_max > max_item_size`.
- Si `max_item_size modulo slab_chunk_max != 0`.
- Si `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

  La valeur `(max_cache_memory - memcached_connections_overhead)` est la mémoire du nœud utilisable pour les données. Pour de plus amples informations, veuillez consulter Surcharge de la connexion Memcached.

Moteurs et versions pris en charge

ElastiCache les caches sans serveur prennent en charge ElastiCache la version 7.2 pour Valkey et versions ultérieures, la ElastiCache version 1.6 pour Memcached et versions ultérieures, et ElastiCache 7.0 pour Redis OSS et versions ultérieures.

ElastiCache les caches conçus par nos soins prennent en charge ElastiCache la version 7.2 pour Valkey et versions ultérieures, la ElastiCache version 1.4.5 pour Memcached et versions ultérieures, et ElastiCache 4.0.10 pour Redis OSS et versions ultérieures.

ElastiCache Les clusters conçus par nos soins prennent en charge les versions de Valkey suivantes :

- Versions de Valkey prises en charge
- ElastiCache version 8.1 pour Valkey
- ElastiCache version 8.0 pour Valkey
- ElastiCache version 7.2.6 pour Valkey

Versions de Valkey prises en charge

Versions de Valkey prises en charge ci-dessous. Notez que Valkey prend en charge la plupart des fonctionnalités disponibles dans ElastiCache la version 7.2 pour Redis OSS par défaut.

- Vous pouvez également mettre à niveau vos ElastiCache clusters avec des versions antérieures à 5.0.6. Le processus impliqué est le même, mais peut entraîner un temps de basculement plus long pendant la propagation DNS (30 s-1 mn).
- À partir de Redis OSS 7, ElastiCache permet de basculer entre Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé).
- Le processus de mise à niveau du moteur Amazon ElastiCache for Redis OSS est conçu pour faire de son mieux pour conserver vos données existantes et nécessite une réplication Redis OSS réussie.
- Lors de la mise à niveau du moteur, les connexions client existantes ElastiCache seront interrompues. Pour minimiser les temps d'arrêt lors des mises à niveau du moteur, nous vous recommandons de mettre en œuvre les meilleures pratiques pour les clients Redis OSS, avec des tentatives d'erreur et des retards exponentiels, ainsi que les meilleures pratiques pour minimiser les temps d'arrêt pendant la maintenance.
- Vous ne pouvez pas passer directement de Valkey ou Redis OSS (mode cluster désactivé) vers Valkey ou Redis OSS (mode cluster activé) lorsque vous mettez à niveau votre moteur. La

procédure suivante explique comment passer de Valkey ou Redis OSS (mode cluster désactivé) vers Valkey ou Redis OSS (mode cluster activé).

Pour passer d'une version de moteur Valkey ou Redis OSS (mode cluster désactivé) vers une version de moteur Valkey ou Redis OSS (mode cluster activé)

1. Effectuez une sauvegarde de votre cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour de plus amples informations, veuillez consulter Réalisation de sauvegardes manuelles.
2. Utilisez la sauvegarde pour créer et amorcer un cluster Valkey ou Redis OSS (mode cluster activé) avec une partition (groupe de nœuds). Spécifiez la nouvelle version du moteur et activez le mode de cluster lors de la création du cluster ou du groupe de réplication. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.
3. Supprimez l'ancien cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour plus d'informations, consultez Supprimer un cluster dans ElastiCache ou Suppression d'un groupe de réplication.
4. Adaptez le nouveau cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster activé) au nombre de partitions (groupes de nœuds) dont vous avez besoin. Pour de plus amples informations, consultez Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).

- Lors de la mise à niveau des versions majeures du moteur, par exemple de 5.0.6 à 6.0, vous devez également choisir un nouveau groupe de paramètres compatible avec la nouvelle version du moteur.
- Pour les clusters Redis OSS uniques et les clusters dont le mode multi-AZ est désactivé, nous recommandons de mettre suffisamment de mémoire à la disposition de Redis OSS, comme décrit dans. S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS Dans ce cas, le réplica principal n'est pas disponible pour traiter les demandes de service pendant la mise à niveau.
- Pour les clusters Redis OSS sur lesquels le mode multi-AZ est activé, nous vous recommandons également de planifier les mises à niveau du moteur pendant les périodes de faible trafic d'écriture entrant. Lors de la mise à niveau vers Redis OSS 5.0.6 ou une version ultérieure, le cluster principal reste disponible pour les demandes de service pendant le processus de mise à niveau.

Les clusters et les groupes de réplication avec plusieurs partitions sont traités et soumis à des correctifs comme suit :

  - Toutes les partitions sont traitées en parallèle. Une seule opération de mise à niveau à la fois est effectuée sur une partition.
  - 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.
- Si les chiffrements sont activés sur votre cluster ou votre groupe de réplication actuel, vous ne pouvez pas effectuer de mise à niveau vers une version du moteur ne prenant pas en charge le chiffrement, comme par exemple de 3.2.6 vers 3.2.10.

Considérations relatives à Memcached

Lorsque vous mettez à niveau un cluster Memcached conçu par vos soins, tenez compte des points suivants.

- La gestion de la version du moteur est conçue afin que vous ayez autant de contrôle que possible sur le déroulement de la correction. Toutefois, ElastiCache se réserve 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 de cache.
- Comme le moteur Memcached ne prend pas en charge la persistance, les mises à niveau de version du moteur Memcached sont toujours un processus perturbateur qui efface toutes les données de cache dans le cluster.

ElastiCache version 8.1 pour Valkey

Voici quelques-unes des nouvelles fonctionnalités introduites dans Valkey 8.1 (par rapport à ElastiCache Valkey 8.0) :

- Une nouvelle implémentation de table de hachage qui réduit la surcharge mémoire afin de réduire l'utilisation de la mémoire jusqu'à 20 % pour les key/value modèles courants.
- Support natif des filtres Bloom, un nouveau type de données qui vous permet d'effectuer des recherches en utilisant jusqu'à 98 % de mémoire en moins par rapport au type de données Set.
- Nouvelle commande COMMANDLOG qui enregistre les exécutions lentes, les demandes volumineuses et les réponses volumineuses.
- Nouveau support de mise à jour conditionnelle de la commande SET à l'aide de l'argument IFEQ.

- Améliorations des performances, notamment une latence jusqu'à 45 % inférieure pour la commande ZRANK, des performances jusqu'à 12 fois plus rapides pour PFMERGE et PFCOUNT, et un débit jusqu'à 514 % plus élevé pour BITCOUNT.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la version Valkey 8.1, consultez les notes de mise à jour de Valkey 8.1

ElastiCache version 8.0 pour Valkey

Voici quelques-unes des nouvelles fonctionnalités introduites dans Valkey 8.0 (par rapport à ElastiCache Valkey 7.2.6) :

- Améliorations de l'efficacité de la mémoire, permettant aux utilisateurs de stocker jusqu'à 20 % de données supplémentaires par nœud sans aucune modification de l'application.
- Infrastructure de métriques par emplacement récemment introduite pour les caches conçus par vos soins, offrant une visibilité détaillée sur les performances et l'utilisation des ressources des emplacements individuels.
- ElastiCache Serverless for Valkey 8.0 peut doubler le nombre de demandes prises en charge par seconde (RPS) toutes les 2 à 3 minutes, pour atteindre 5 millions de RPS par cache contre zéro en moins de 13 minutes, avec une latence de lecture constante inférieure à la milliseconde p50.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la version de Valkey 8, consultez les notes de mise à jour de Valkey 8

ElastiCache version 7.2.6 pour Valkey

Le 10 octobre 2024, ElastiCache la version 7.2.6 pour Valkey est sortie. Voici quelques-unes des nouvelles fonctionnalités introduites dans la version 7.2 (par rapport à la ElastiCache version 7.1 pour Redis OSS) :

- Optimisations des performances et de la mémoire pour différents types de données : optimisation de la mémoire pour les touches de type liste et ensemble, optimisation de la vitesse pour les commandes d'ensembles triés, optimisation des performances pour les commandes comportant plusieurs touches en mode cluster, amélioration des pub/sub performances, optimisation des performances pour les commandes SCAN, SSCAN, HSCAN, ZSCAN et de nombreuses autres optimisations de moindre envergure.

- Nouvelle option WITHSCORE pour les commandes ZRANK et ZREVRANK
- CLIENT NO-TOUCH pour que les clients puissent exécuter des commandes sans affecter LRU/LFU les touches.
- Nouvelle commande CLUSTER MYSHARDID qui renvoie l'ID de partition du nœud pour regrouper logiquement les nœuds en mode cluster en fonction de la réplication.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la ElastiCache version 7.2 pour Valkey, consultez les notes de mise à jour de Redis OSS 7.2.4 (la ElastiCache version 7.2 pour Valkey inclut toutes les modifications depuis la ElastiCache version 7.1 pour Redis OSS jusqu'à la ElastiCache version 7.2.4 pour Redis OSS). Notes de mise à jour de Valkey 7.2 chez Valkey on. GitHub

ElastiCache version 8.1 pour Valkey

Voici quelques-unes des nouvelles fonctionnalités introduites dans Valkey 8.1 (par rapport à ElastiCache Valkey 8.0) :

- Une nouvelle implémentation de table de hachage qui réduit la surcharge mémoire afin de réduire l'utilisation de la mémoire jusqu'à 20 % pour les key/value modèles courants.
- Support natif des filtres Bloom, un nouveau type de données qui vous permet d'effectuer des recherches en utilisant jusqu'à 98 % de mémoire en moins par rapport au type de données Set.
- Nouvelle commande COMMANDLOG qui enregistre les exécutions lentes, les demandes volumineuses et les réponses volumineuses.
- Nouveau support de mise à jour conditionnelle de la commande SET à l'aide de l'argument IFEQ.
- Améliorations des performances, notamment une latence jusqu'à 45 % inférieure pour la commande ZRANK, des performances jusqu'à 12 fois plus rapides pour PFMERGE et PFCOUNT, et un débit jusqu'à 514 % plus élevé pour BITCOUNT.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la version Valkey 8.1, consultez les notes de mise à jour de Valkey 8.1

ElastiCache version 8.0 pour Valkey

Voici quelques-unes des nouvelles fonctionnalités introduites dans Valkey 8.0 (par rapport à ElastiCache Valkey 7.2.6) :

- Améliorations de l'efficacité de la mémoire, permettant aux utilisateurs de stocker jusqu'à 20 % de données supplémentaires par nœud sans aucune modification de l'application.
- Infrastructure de métriques par emplacement récemment introduite pour les caches conçus par vos soins, offrant une visibilité détaillée sur les performances et l'utilisation des ressources des emplacements individuels.
- ElastiCache Serverless for Valkey 8.0 peut doubler le nombre de demandes prises en charge par seconde (RPS) toutes les 2 à 3 minutes, pour atteindre 5 millions de RPS par cache contre zéro en moins de 13 minutes, avec une latence de lecture constante inférieure à la milliseconde p50.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la version de Valkey 8, consultez les notes de mise à jour de Valkey 8

ElastiCache version 7.2.6 pour Valkey

Le 10 octobre 2024, ElastiCache la version 7.2.6 pour Valkey est sortie. Voici quelques-unes des nouvelles fonctionnalités introduites dans la version 7.2 (par rapport à la ElastiCache version 7.1 pour Redis OSS) :

- Optimisations des performances et de la mémoire pour différents types de données : optimisation de la mémoire pour les touches de type liste et ensemble, optimisation de la vitesse pour les commandes d'ensembles triés, optimisation des performances pour les commandes comportant plusieurs touches en mode cluster, amélioration des pub/sub performances, optimisation des performances pour les commandes SCAN, SSCAN, HSCAN, ZSCAN et de nombreuses autres optimisations de moindre envergure.
- Nouvelle option WITHSCORE pour les commandes ZRANK et ZREVRANK
- CLIENT NO-TOUCH pour que les clients puissent exécuter des commandes sans affecter LRU/LFU les touches.
- Nouvelle commande CLUSTER MYSHARDID qui renvoie l'ID de partition du nœud pour regrouper logiquement les nœuds en mode cluster en fonction de la réplication.

Pour plus d'informations sur Valkey, voir Valkey

Pour plus d'informations sur la ElastiCache version 7.2 pour Valkey, consultez les notes de mise à jour de Redis OSS 7.2.4 (la ElastiCache version 7.2 pour Valkey inclut toutes les modifications depuis la ElastiCache version 7.1 pour Redis OSS jusqu'à la ElastiCache version 7.2.4 pour Redis OSS). Notes de mise à jour de Valkey 7.2 chez Valkey on. GitHub

Versions du moteur Redis OSS prises en charge

ElastiCache Les caches sans serveur et les caches conçus par nos soins prennent en charge toutes les versions 7.1 et antérieures de Redis OSS.

- ElastiCache version 7.1 pour Redis OSS (améliorée)

ElastiCache Les clusters conçus par nos soins prennent en charge les versions Redis OSS suivantes :

- ElastiCache version 7.1 pour Redis OSS (améliorée)
- ElastiCache version 7.0 pour Redis OSS (améliorée)
- ElastiCache version 6.2 pour Redis OSS (améliorée)
- ElastiCache version 6.0 pour Redis OSS (améliorée)
- ElastiCache version 5.0.6 pour Redis OSS (améliorée)
- ElastiCache version 5.0.5 pour Redis OSS (obsolète, utilisez la version 5.0.6)
- ElastiCache version 5.0.4 pour Redis OSS (obsolète, utilisez la version 5.0.6)
- ElastiCache version 5.0.3 pour Redis OSS (obsolète, utilisez la version 5.0.6)
- ElastiCache version 5.0.0 pour Redis OSS (obsolète, utilisez la version 5.0.6)
- ElastiCache version 4.0.10 pour Redis OSS (améliorée)
- Versions dont la fin de vie (EOL) est passée (3.x)
- Versions dont la fin de vie (EOL) est passée (2.x)

ElastiCache version 7.1 pour Redis OSS (améliorée)

Cette version contient des améliorations de performances qui permettent aux charges de travail d'augmenter le débit et de réduire les latences opérationnelles. ElastiCache la version 7.1 pour Redis OSS introduit deux améliorations principales :

Nous avons étendu la fonctionnalité améliorée I/O des threads afin de gérer également la logique de la couche de présentation. Par couche de présentation, nous entendons les threads d'E/S améliorés qui non seulement lisent les entrées du client, mais les analysent également au format de commande binaire Redis OSS. Elles sont ensuite transférées au thread principal pour exécution, ce qui permet un gain de performance. Modèle d'accès à la mémoire Redis OSS amélioré. Les étapes d'exécution de nombreuses opérations de structure de données sont entrelacées afin de garantir un accès parallèle à la mémoire et de réduire la latence d'accès à la mémoire. Lorsqu'ils utilisent un système

basé ElastiCache sur Graviton3 `R7g.4xlarge` ou une version supérieure, les clients peuvent traiter plus d'un million de requêtes par seconde et par nœud. Grâce aux améliorations de performances apportées à ElastiCache Redis OSS v7.1, les clients peuvent obtenir jusqu'à 100 % de débit en plus et 50 % de latence P99 en moins par rapport à ElastiCache Redis OSS v7.0. Ces améliorations sont activées sur les nœuds dotés d'au moins 8 cœurs physiques (`2xlarge` sur Graviton et `4xlarge` sur x86), quel que soit le type de CPU, et ne nécessitent aucune modification du client.

> **Note**
>
> ElastiCache La version 7.1 est compatible avec Redis OSS v7.0.

ElastiCache version 7.0 pour Redis OSS (améliorée)

ElastiCache pour Redis OSS 7.0 ajoute un certain nombre d'améliorations et prend en charge de nouvelles fonctionnalités :

- Fonctions : ElastiCache pour Redis OSS, 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 ElastiCache cluster, sans obliger les clients à renvoyer les scripts au serveur à chaque connexion.
- Améliorations de l'ACL : Valkey et Redis OSS 7 ajoutent le support pour la prochaine version de Access Control Lists ()ACLs. Les clients peuvent désormais spécifier plusieurs ensembles d'autorisations sur des clés ou des espaces de touches spécifiques dans Valkey et Redis OSS.
- Sharded Pub/Sub : ElastiCache pour Valkey et Redis, OSS 7 ajoute la prise en charge des Pub/Sub functionality in a sharded way when running ElastiCache in Cluster Mode Enabled (CME). Pub/Sub fonctionnalités permettant aux éditeurs d'envoyer des messages à n'importe quel nombre d'abonnés sur une chaîne. Les canaux sont liés à une partition du ElastiCache cluster, ce qui élimine le besoin de propager les informations des canaux entre les partitions, ce qui améliore l'évolutivité.
- I/O Multiplexage amélioré : ElastiCache pour Valkey et Redis, OSS 7 introduit un multiplexage d'E/S amélioré, qui fournit un débit accru et une latence réduite pour les charges de travail à haut débit associées à de nombreuses connexions client simultanées à un cluster. ElastiCache Par exemple, lorsque vous utilisez un cluster de nœuds r6g.xlarge et que vous exécutez 5 200 clients simultanés, vous pouvez obtenir jusqu'à 72 % d'augmentation du débit (opérations de lecture et d'écriture par seconde) et une diminution de 71 % de la latence P99, par rapport à la version 6 pour Redis OSS. ElastiCache

Pour plus d'informations sur Valkey, voir Valkey. 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

ElastiCache version 6.2 pour Redis OSS (améliorée)

ElastiCache pour Redis OSS 6.2 inclut des améliorations de performances pour les clusters compatibles TLS utilisant des types de nœuds x86 avec 8 V CPUs ou plus ou des types de nœuds Graviton2 avec 4 V ou plus. CPUs Ces améliorations améliorent le débit et réduisent le temps d'établissement de la connexion client en transférant le chiffrement vers d'autres versions. CPUs Avec Redis OSS 6.2, vous pouvez également gérer l'accès aux canaux Pub/Sub à l'aide de règles de liste de contrôle d'accès (ACL).

Avec cette version, nous introduisons également la prise en charge de la hiérarchisation des données sur les nœuds de cluster contenant un NVMe SSD connecté localement. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

La version 6.2.6 du moteur Redis OSS 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 de Redis OSS APIs pour les applications qui fonctionnent via JSON. Pour plus d'informations, consultez Mise en route avec JSON. Sont également incluses les métriques liées au JSON `JsonBasedCmds` et`JsonBasedCmdsLatency`, qui sont intégrées CloudWatch pour surveiller l'utilisation de ce type de données. Pour de plus amples informations, veuillez consulter Métriques pour Valkey et Redis OSS.

Vous spécifiez la version du moteur à l'aide de la version 6.2. ElastiCache invoquera automatiquement la version de correctif préférée de Redis OSS 6.2 disponible. Par exemple, lorsque vous créez/modifiez un cluster de cache, vous définissez le paramètre `--engine-version` sur 6.2. Le cluster sera lancé avec la version de correctif préférée actuellement disponible de Redis OSS 6.2 à ce creation/modification moment-là. Si vous spécifiez la version 6.x du moteur dans l'API, vous obtiendrez la dernière version mineure de Redis OSS 6.

Pour les clusters 6.0 existants, vous pouvez opter pour la prochaine mise à niveau automatique de la version mineure en définissant le `AutoMinorVersionUpgrade` paramètre sur `yes` dans le `CreateCacheClusterModifyCacheCluster`, `CreateReplicationGroup` ou `ModifyReplicationGroup` APIs. ElastiCache mettra à niveau la version mineure de vos clusters 6.0 existants vers la version 6.2 à l'aide de mises à jour en libre-service. Pour plus d'informations, consultez la section Mises à jour en libre-service sur Amazon ElastiCache.

Lors de l'appel de l' DescribeCacheEngineVersions API, la valeur du `EngineVersion` paramètre sera définie sur 6.2 et la version réelle du moteur avec la version du correctif sera renvoyée dans le `CacheEngineVersionDescription` champ.

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

ElastiCache version 6.0 pour Redis OSS (améliorée)

Amazon ElastiCache présente la prochaine version du ElastiCache moteur Redis OSS, qui inclut l'authentification des utilisateurs grâce au contrôle d'accès basé sur les rôles, la mise en cache côté client et des améliorations opérationnelles significatives.

À partir de Redis OSS 6.0, nous ElastiCache proposerons une version unique pour chaque version mineure de Redis OSS, au lieu de proposer plusieurs versions de correctif. ElastiCache gérera automatiquement la version du correctif de vos clusters de cache en cours d'exécution, garantissant ainsi de meilleures performances et une sécurité renforcée.

Vous pouvez également opter pour la prochaine mise à niveau automatique de la version mineure en définissant le `AutoMinorVersionUpgrade` paramètre sur `yes` et vous ElastiCache gérerez la mise à niveau de la version mineure par le biais de mises à jour en libre-service. Pour de plus amples informations, veuillez consulter Mises à jour du service dans ElastiCache.

Vous spécifiez la version du moteur en utilisant`6.0`. ElastiCache invoquera automatiquement la version de correctif préférée de Redis OSS 6.0 disponible. Par exemple, lorsque vous êtes create/modify un cluster de cache, vous définissez le `--engine-version` paramètre sur 6.0. Le cluster sera lancé avec la version de correctif préférée actuellement disponible de Redis OSS 6.0 au moment de la création/modification. Toute demande avec une valeur de version de correctif spécifique sera rejetée, une exception sera levée et le processus échouera.

Lors de l'appel de l' DescribeCacheEngineVersions API, la valeur du `EngineVersion` paramètre sera définie sur 6.0 et la version réelle du moteur avec la version du correctif sera renvoyée dans le `CacheEngineVersionDescription` champ.

Pour plus d'informations sur la version 6.0 de Redis OSS, consultez les notes de mise à jour de Redis OSS 6.0 sur Redis OSS on. GitHub

ElastiCache version 5.0.6 pour Redis OSS (améliorée)

Amazon ElastiCache présente la prochaine version du ElastiCache moteur Redis OSS, qui inclut des corrections de bogues et les mises à jour cumulatives suivantes :

- Garantie de stabilité du moteur dans des conditions particulières
- Amélioration de la gestion des erreurs Hyperloglog
- Amélioration des commandes de liaison pour une réplication fiable
- Suivi continu de la diffusion des messages via la commande `XCLAIM`
- Amélioration de la gestion du champ `LFU` dans les objets
- Amélioration de la gestion des transactions en cas d'utilisation de `ZPOP`
- Possibilité de renommer des commandes : paramètre appelé `rename-commands` qui vous permet de renommer des commandes Redis OSS potentiellement dangereuses ou coûteuses susceptibles d'entraîner une perte de données accidentelle, telle que ou. `FLUSHALL` `FLUSHDB` Ceci est similaire à la configuration de la commande de changement de nom dans le logiciel libre Redis OSS. Cependant, ElastiCache a amélioré l'expérience en fournissant un flux de travail entièrement géré. Les modifications des noms de commande sont appliquées immédiatement et propagées automatiquement sur tous les nœuds du cluster qui contiennent la liste de commandes. Vous n'avez pas besoin d'intervenir, par exemple en redémarrant les nœuds.

  Les exemples suivants montrent comment modifier les groupes de paramètres existants. Ils incluent le paramètre `rename-commands`, qui est une liste de commandes séparées par des espaces que vous souhaitez renommer.

  ```
  aws elasticache modify-cache-parameter-group --cache-parameter-group-
  name custom_param_group
  --parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
   restrictedflushall'" --region region
  ```

  Dans cet exemple, le paramètre rename-commands est utilisé pour renommer la commande `flushall` en `restrictedflushall`.

  Pour renommer plusieurs commandes, utilisez ce qui suit :

  ```
  aws elasticache modify-cache-parameter-group --cache-parameter-group-
  name custom_param_group
  --parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
   restrictedflushall flushdb restrictedflushdb'" --region region
  ```

  Pour annuler une modification, réexécutez la commande renommée et excluez les valeurs renommées de la liste `ParameterValue` que vous souhaitez conserver, comme illustré ci-après :

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-
name custom_param_group
--parameter-name-values "ParameterName=rename-commands,  ParameterValue='flushall
 restrictedflushall'" --region region
```

Dans ce cas, la commande `flushall` est renommée `restrictedflushall` et toutes les autres commandes renommées reprennent leur nom de commande initial.

> **Note**
>
> Lorsque vous renommez des commandes, vous devez respecter les limitations suivantes :
>
> - Toutes les commandes renommées doivent être alphanumériques.
> - La longueur maximale de nouveaux noms de commande est de 20 caractères alphanumériques.
> - Lorsque vous renommez des commandes, assurez-vous que vous mettez à jour le groupe de paramètres associé à votre cluster.
> - Pour empêcher une utilisation entière de la commande, utilisez le mot clé `blocked`, comme illustré ci-après :
>
> ```
> aws elasticache modify-cache-parameter-group --cache-parameter-group-
> name custom_param_group
> --parameter-name-values "ParameterName=rename-commands,
>  ParameterValue='flushall blocked'" --region region
> ```

Pour plus d'informations sur les modifications de paramètre et une liste des commandes qui sont éligibles pour renommer, consultez Modifications des paramètres de Redis OSS 5.0.3.

- Redis OSS Streams : Cela modélise une structure de données de journal qui permet aux producteurs d'ajouter de nouveaux éléments en temps réel. Il permet également aux consommateurs d'utiliser des messages, de façon bloquante ou non. Les flux permettent aussi aux groupes de consommateurs, qui représentent un groupe de clients, d'utiliser en collaboration différentes parties du même flux de messages, comme avec Apache Kafka. Pour plus d'informations, consultez Streams.
- Prise en charge d'une famille de commandes de flux, telles que `XADD`, `XRANGE` et `XREAD`. Pour plus d'informations, consultez la section Commandes Streams.

- Plusieurs paramètres nouveaux et renommés. Pour de plus amples informations, veuillez consulter Modifications des paramètres de Redis OSS 5.0.0.
- Une nouvelle métrique Redis OSS,`StreamBasedCmds`.
- Temps de capture légèrement plus rapide pour les nœuds Redis OSS.

> **Important**
>
> ElastiCache a rétroporté deux corrections de bogues critiques depuis la version open source 5.0.1 de Redis OSS. Elles sont répertoriées ci-dessous.
>
> - Réponse incohérente de RESTORE lorsque certaines clés ont déjà expiré.
> - La commande `XCLAIM` peut éventuellement renvoyer une entrée incorrecte ou désynchroniser le protocole.
>
> Ces deux corrections de bogues sont incluses dans le support de Redis OSS ElastiCache pour la version 5.0.0 du moteur Redis OSS et seront prises en compte dans les futures mises à jour de version.

Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 5.0.6 sur Redis OSS on. GitHub

ElastiCache version 5.0.5 pour Redis OSS (obsolète, utilisez la version 5.0.6)

Amazon ElastiCache présente la prochaine version de ElastiCache pour le moteur Redis OSS ;. Il inclut les modifications de configuration en ligne pour les clusters ElastiCache de basculement automatique au cours de toutes les opérations planifiées. Vous pouvez désormais faire évoluer votre cluster, mettre à niveau la version du moteur Redis OSS et appliquer des correctifs et des mises à jour de maintenance pendant que le cluster reste en ligne et continue de traiter les demandes entrantes. Elle inclut également des correctifs de bogue.

Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 5.0.5 sur Redis OSS on. GitHub

ElastiCache version 5.0.4 pour Redis OSS (obsolète, utilisez la version 5.0.6)

Amazon ElastiCache présente la prochaine version du moteur Redis OSS prise en charge par ElastiCache. Elle inclut les améliorations suivantes :

- Garantie de stabilité du moteur dans des conditions particulières
- Amélioration de la gestion des erreurs Hyperloglog
- Amélioration des commandes de liaison pour une réplication fiable
- Suivi continu de la diffusion des messages via la commande `XCLAIM`
- Amélioration de la gestion du champ `LFU` dans les objets
- Amélioration de la gestion des transactions en cas d'utilisation de `ZPOP`

Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 5.0.4 sur Redis OSS on. GitHub

ElastiCache version 5.0.3 pour Redis OSS (obsolète, utilisez la version 5.0.6)

Amazon ElastiCache présente la prochaine version du ElastiCache moteur Redis OSS, qui inclut des corrections de bogues.

ElastiCache version 5.0.0 pour Redis OSS (obsolète, utilisez la version 5.0.6)

Amazon ElastiCache présente la prochaine version majeure de ElastiCache pour le moteur Redis OSS. ElastiCache la version 5.0.0 pour Redis OSS prend en charge les améliorations suivantes :

- Redis OSS Streams : Cela modélise une structure de données de journal qui permet aux producteurs d'ajouter de nouveaux éléments en temps réel. Il permet également aux consommateurs d'utiliser des messages, de façon bloquante ou non. Les flux permettent aussi aux groupes de consommateurs, qui représentent un groupe de clients, d'utiliser en collaboration différentes parties du même flux de messages, comme avec Apache Kafka. Pour plus d'informations, consultez Streams.
- Prise en charge d'une famille de commandes de flux, telles que `XADD`, `XRANGE` et `XREAD`. Pour plus d'informations, consultez la section Commandes Streams.
- Plusieurs paramètres nouveaux et renommés. Pour de plus amples informations, veuillez consulter Modifications des paramètres de Redis OSS 5.0.0.
- Une nouvelle métrique Redis OSS,`StreamBasedCmds`.
- Temps de capture légèrement plus rapide pour les nœuds Redis OSS.

ElastiCache version 4.0.10 pour Redis OSS (améliorée)

Amazon ElastiCache présente la prochaine version majeure de ElastiCache pour le moteur Redis OSS. ElastiCache la version 4.0.10 pour Redis OSS prend en charge les améliorations suivantes :

- Le redimensionnement du cluster en ligne et le chiffrement en une seule ElastiCache version. Pour plus d'informations, consultez les ressources suivantes :
  - Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)
  - Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)
  - Sécurité des données sur Amazon ElastiCache
- Plusieurs nouveaux paramètres. Pour de plus amples informations, veuillez consulter Modifications des paramètres de Redis OSS 4.0.10.
- Prise en charge des familles de commandes de mémoire, comme `MEMORY`. Pour plus d'informations, voir Commandes (recherche dans MEMO).
- Prise en charge de la défragmentation de mémoire en ligne, tout en permettant une utilisation plus efficace de la mémoire et plus de mémoire disponible pour vos données.
- Support pour les purges et les suppressions asynchrones. ElastiCache pour Redis OSS prend en charge des commandes telles que`UNLINK`, `FLUSHDB` et `FLUSHALL` pour s'exécuter dans un thread différent du thread principal. Cette étape aide à améliorer les performances et les temps de réponse pour vos applications en libérant de la mémoire de manière asynchrone.
- Une nouvelle métrique Redis OSS,`ActiveDefragHits`. Pour plus d'informations, consultez Metrics for Redis OSS.

Les utilisateurs de Redis OSS (mode cluster désactivé) exécutant la ElastiCache version 3.2.10 pour Redis OSS peuvent utiliser la console pour mettre à niveau leurs clusters via une mise à niveau en ligne.

Comparaison ElastiCache du redimensionnement des clusters et de la prise en charge du chiffrement

Fonctionnalité	3.2.6	3.2.10	4.0.10 et ultérieures
Redimensionnement de cluster en ligne *	Non	Oui	Oui
Chiffrement en transit **	Oui	Non	Oui
Chiffrement au repos **	Oui	Non	Oui

Fonctionnalité	3.2.6	3.2.10	4.0.10 et ultérieures
* Ajout, suppression et rééquilibrage de partitions.			
** Requis pour les applications conformes aux réglementations FedRAMP, HIPAA et PCI DSS. Pour de plus amples informations, veuillez consulter Validation de conformité pour Amazon ElastiCache.			

Versions dont la fin de vie (EOL) est passée (3.x)

ElastiCache version 3.2.10 pour Redis OSS (améliorée)

Amazon ElastiCache présente la prochaine version majeure de ElastiCache pour le moteur Redis OSS. ElastiCache la version 3.2.10 pour Redis OSS (améliorée) introduit le redimensionnement du cluster en ligne pour ajouter ou supprimer des fragments du cluster tout en continuant à traiter les demandes entrantes. I/O ElastiCache pour Redis OSS 3.2.10, les utilisateurs disposent de toutes les fonctionnalités des versions antérieures de Redis OSS, à l'exception de la possibilité de chiffrer leurs données. Cette fonctionnalité est actuellement disponible pour la version 3.2.6 uniquement.

Comparaison des ElastiCache versions 3.2.6 et 3.2.10 pour Redis OSS

Fonctionnalité	3.2.6	3.2.10
Redimensionnement de cluster en ligne *	Non	Oui
Chiffrement en transit **	Oui	Non
Chiffrement au repos **	Oui	Non
* Ajout, suppression et rééquilibrage de partitions.		
** Requis pour les applications conformes aux réglementations FedRAMP, HIPAA et PCI DSS. Pour de plus amples informations, veuillez consulter Validation de conformité pour Amazon ElastiCache.		

Pour plus d'informations, consultez les ressources suivantes :

- Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)
- Redimensionnement de cluster en ligne

ElastiCache version 3.2.6 pour Redis OSS (améliorée)

Amazon ElastiCache présente la prochaine version majeure de ElastiCache pour le moteur Redis OSS. ElastiCache version 3.2.6 pour les utilisateurs de Redis OSS ont accès à toutes les fonctionnalités des versions antérieures de Redis OSS, ainsi qu'à la possibilité de chiffrer leurs données. Pour plus d'informations, consultez les ressources suivantes :

- ElastiCache chiffrement en transit (TLS)
- Chiffrement au repos dans ElastiCache
- Validation de conformité pour Amazon ElastiCache

ElastiCache version 3.2.4 pour Redis OSS (améliorée)

ElastiCache La version 3.2.4 d'Amazon introduit la prochaine version majeure de ElastiCache pour le moteur Redis OSS. ElastiCache 3.2.4 Les utilisateurs disposent de toutes les fonctionnalités des versions antérieures de Redis OSS, ainsi que de la possibilité d'exécuter en mode cluster ou en mode hors cluster. Le tableau suivant résume les possibilités.

Comparaison du mode hors cluster et du mode cluster de Redis OSS 3.2.4

Fonctionnalité	Mode non-cluster	Mode cluster
Partitionnement de données	Non	Oui
Indexation géospatiale	Oui	Oui
Modifier le type de nœud	Oui	Oui *
Développement de réplica	Oui	Oui *
Monter en puissance	Non	Oui *
Prise en charge de base de données	Plusieurs	Unique

Fonctionnalité	Mode non-cluster	Mode cluster
Groupe de paramètres	`default.redis3.2` **	`default.redis3.2.cluster.on` **

* Voir Restauration à partir d'une sauvegarde dans un nouveau cache

** Ou un dérivé de celui-ci.

Remarques :

- Partitionnement : possibilité de fractionner vos données sur 2 à 500 groupes de nœuds (partitions) avec prise en charge de la réplication pour chaque groupe de nœuds.
- Indexation géospatiale — Redis OSS 3.2.4 introduit la prise en charge de l'indexation géospatiale via six commandes GEO. Pour plus d'informations, consultez la documentation des commandes Redis OSS GEO* Commandes : GEO sur la page des commandes Valkey (filtrée pour GEO).

Pour plus d'informations sur les fonctionnalités supplémentaires de Redis OSS 3, consultez les notes de version de Redis OSS 3.2 et les notes de version de Redis OSS 3.0.

Valkey ou Redis OSS actuellement ElastiCache géré (mode cluster activé) ne prend pas en charge les fonctionnalités Redis OSS 3.2 suivantes :

- Migration de réplica
- Rééquilibrage du cluster
- Débogueur LUA

ElastiCache désactive les commandes de gestion Redis OSS 3.2 suivantes :

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`

- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumpepoch`
- `cluster set-config-epoch`
- `cluster reset`

Pour plus d'informations sur les paramètres de Redis OSS 3.2.4, consultez. Modifications des paramètres de Redis OSS 3.2.4

Versions dont la fin de vie (EOL) est passée (2.x)

ElastiCache version 2.8.24 pour Redis OSS (améliorée)

Les améliorations apportées à Redis OSS depuis la version 2.8.23 incluent des corrections de bogues et l'enregistrement des mauvaises adresses d'accès à la mémoire. Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 2.8.

ElastiCache version 2.8.23 pour Redis OSS (améliorée)

Les améliorations apportées à Redis OSS depuis la version 2.8.22 incluent des corrections de bogues. Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 2.8. Cette version comprend également la prise en charge du nouveau paramètre `close-on-slave-write` qui, s'il est activé, déconnecte les clients qui tentent d'écrire sur un réplica en lecture seule.

Pour plus d'informations sur les paramètres de Redis OSS 2.8.23, consultez le guide de Paramètres ajoutés à Redis OSS 2.8.23 (amélioré) l' ElastiCache utilisateur.

ElastiCache version 2.8.22 pour Redis OSS (améliorée)

Les améliorations apportées à Redis OSS depuis la version 2.8.21 sont les suivantes :

- Prise en charge des synchronisations et sauvegardes sans fonction fork, ce qui vous permet d'attribuer moins de mémoire pour les sauvegardes et plus de mémoire pour votre application. Pour de plus amples informations, veuillez consulter Implémentation de la sauvegarde et de la synchronisation. Le processus sans fonction fork peut avoir un impact sur la latence et le débit. Dans le cas d'un débit d'écriture élevé, lorsqu'un réplica relance une synchronisation, il peut être inaccessible pendant toute la durée de la synchronisation.

- En cas de basculement, les groupes de réplication récupèrent désormais plus rapidement, car les réplicas effectuent des synchronisations partielles avec le réplica principal plutôt que des synchronisations complètes, dans la mesure du possible. En outre, les réplicas principaux et les réplicas en lecture n'utilisent plus le disque pendant les synchronisations, permettant ainsi d'optimiser la vitesse.
- Support pour deux nouvelles CloudWatch métriques.
  - `ReplicationBytes` : le nombre d'octets qu'un cluster principal du groupe de réplication envoie aux réplicas en lecture.
  - `SaveInProgress` : une valeur binaire qui indique si un processus de sauvegarde en arrière-plan est en cours d'exécution.

  Pour de plus amples informations, veuillez consulter Surveillance de l'utilisation à l'aide de CloudWatch métriques.
- Plusieurs correctifs de bogues essentiels dans le comportement de réplication PSYNC. Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 2.8.
- Afin de maintenir des performances de réplication améliorées dans les groupes de réplication multi-AZ et d'améliorer la stabilité du cluster, les ElastiCache non-répliques ne sont plus prises en charge.
- Pour améliorer la cohérence des données entre le cluster principal et les réplicas d'un groupe de réplication, les réplicas ne supprimeront plus les clés indépendantes du cluster principal.
- Les variables de configuration Redis OSS ne `appendfsync` sont pas prises en charge sur Redis OSS version 2.8.22 `appendonly` et versions ultérieures.
- En cas de mémoire insuffisante, les clients avec un important tampon de sortie peuvent être déconnectés d'un cluster de réplica. En cas de déconnexion, le client doit se reconnecter. Ces situations sont plus susceptibles de se produire pour les clients PUBSUB.

ElastiCache version 2.8.21 pour Redis OSS

Les améliorations apportées à Redis OSS depuis la version 2.8.19 incluent un certain nombre de corrections de bogues. Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 2.8.

ElastiCache version 2.8.19 pour Redis OSS

Les améliorations apportées à Redis OSS depuis la version 2.8.6 sont les suivantes :

- Support pour HyperLogLog. Pour plus d'informations, consultez la nouvelle structure de données de Redis OSS : HyperLogLog.
- Le type de données « ensemble trié » prend désormais en charge les requêtes de plage lexicographiques avec les nouvelles commandes `ZRANGEBYLEX`, `ZLEXCOUNT` et `ZREMRANGEBYLEX`.
- Pour éviter qu'un nœud principal envoie des données obsolètes aux nœuds de réplica, la synchronisation principale échoue si un processus enfant (`bgsave`) d'enregistrement en arrière-plan est interrompu.
- Support pour la HyperLogLogBasedCommands CloudWatchmétrique. Pour de plus amples informations, veuillez consulter Métriques pour Valkey et Redis OSS.

ElastiCache version 2.8.6 pour Redis OSS

Les améliorations apportées à Redis OSS depuis la version 2.6.13 sont les suivantes :

- Résilience et tolérance aux pannes optimisées pour les réplicas en lecture.
- Prise en charge de la resynchronisation partielle.
- Prise en charge du nombre minimum de réplicas en lecture, défini par l'utilisateur, qui doit être disponible en permanence.
- Prise en charge intégrale de la fonction pub/sub — notification aux clients des événements survenus sur le serveur.
- Détection automatique d'une défaillance du nœud principal et basculement de votre nœud principal vers un nœud secondaire.

ElastiCache version 2.6.13 pour Redis OSS

ElastiCache la version 2.6.13 pour Redis OSS était la version initiale supportée par Redis OSS. ElastiCache Le multi-AZ n'est pas pris en charge dans la ElastiCache version 2.6.13 pour Redis OSS.

ElastiCache versions pour le calendrier de fin de vie de Redis OSS

Cette section définit les dates de fin de vie (EOL) pour les anciennes versions majeures au fur et à mesure qu'elles sont annoncées. Cela vous permet de prendre des décisions concernant les versions et les mises à niveau pour l'avenir.

> **Note**
>
> ElastiCache les versions 5.0.0 à 5.0.5 pour Redis OSS sont obsolètes. Utilisez les versions 5.0.6 ou ultérieures.

Le tableau suivant présente le calendrier du Support étendu ElastiCache pour les moteurs Redis OSS.

Support étendu et calendrier de fin de vie

Version principale du moteur	Support standard terminé	Début du Support étendu Y1 Premium	Début du Support étendu Y2 Premium	Début du Support étendu Y3 Premium	Fin du support étendu et de la version EOL
Redis OSS v4	31/01/2026	01/02/2026	01/02/2027	01/02/2028	31/01/2029
Redis OSS v5	31/01/2026	01/02/2026	01/02/2027	01/02/2028	31/01/2029
Redis OSS 6	31/01/2027	01/02/2027	01/02/2028	01/02/2029	31/01/2030

Le tableau suivant récapitule chaque version et sa date EOL annoncée, ainsi que la version cible de mise à niveau recommandée.

EOL passée

V n d la s	Versions mineures de la source	Cible de mise à niveau recommandée	Date EOL
V	3.2.4, 3.2.6 et 3.2.10	Version 6.2 ou supérieure	31 juillet 2023

V n d la s	Versions mineures de la source	Cible de mise à niveau recommandée	Date EOL
		Note Pour les régions US-ISO-EAST US-ISO-WEST -1, -1 et US-ISOB-EAST -1, nous recommandons la version 5.0.6 ou supérieure.	
V	2,8,24, 2,8,23, 2,8,22, 2,8,21, 2.8,19, 2.8,12, 2,8.6, 2.6,13	Version 6.2 ou supérieure Note Pour les régions US-ISO-EAST US-ISO-WEST -1, -1 et US-ISOB-EAST -1, nous recommandons la version 5.0.6 ou supérieure.	13 janvier 2023

Pris en charge ElastiCache pour les versions de Memcached

ElastiCache prend en charge les versions suivantes de Memcached et la mise à niveau vers des versions plus récentes. Dans ce dernier cas, faites attention aux conditions qui causeront l'échec de la mise à niveau si elles ne sont pas remplies.

ElastiCache pour les versions de Memcached

- ElastiCache version 1.6.22 pour Memcached

- ElastiCache version 1.6.17 pour Memcached
- ElastiCache version 1.6.12 pour Memcached
- ElastiCache version 1.6.6 pour Memcached
- ElastiCache version 1.5.16 pour Memcached
- ElastiCache version 1.5.10 pour Memcached
- ElastiCache version 1.4.34 pour Memcached
- ElastiCache version 1.4.33 pour Memcached
- ElastiCache version 1.4.24 pour Memcached
- ElastiCache version 1.4.14 pour Memcached
- ElastiCache version 1.4.5 pour Memcached

ElastiCache version 1.6.22 pour Memcached

ElastiCache pour Memcached version 1.6.22 pour Memcached ajoute le support pour Memcached version 1.6.22. Il n'inclut aucune nouvelle fonctionnalité, mais offre des correctifs de bogue et des mises à jour cumulatives à partir de Memcached 1.6.18.

Pour plus d'informations, consultez ReleaseNotes1622 sur Memcached on. GitHub

ElastiCache version 1.6.17 pour Memcached

ElastiCache pour Memcached version 1.6.17 pour Memcached ajoute le support pour le moteur Memcached version 1.6.17. Il n'inclut aucune nouvelle fonctionnalité, mais offre des correctifs de bogue et des mises à jour cumulatives à partir de Memcached 1.6.17.

Pour plus d'informations, consultez ReleaseNotes1617 sur Memcached on. GitHub

ElastiCache version 1.6.12 pour Memcached

ElastiCache pour Memcached, la version 1.6.12 pour Memcached ajoute la prise en charge du moteur Memcached 1.6.12 et du chiffrement en transit. Il inclut également les correctifs de bogue et les mises à jour cumulatives de Memcached 1.6.6.

Pour plus d'informations, consultez ReleaseNotes1612 sur Memcached on. GitHub

ElastiCache version 1.6.6 pour Memcached

ElastiCache pour Memcached version 1.6.6 pour Memcached ajoute le support pour Memcached version 1.6.6. Il n'inclut aucune nouvelle fonctionnalité, mais inclut des corrections de bogues et des mises à jour cumulatives depuis Memcached 1.5.16. ElastiCache pour Memcached n'inclut pas le support pour Extstore.

Pour plus d'informations, consultez ReleaseNotes166 sur Memcached on. GitHub

ElastiCache version 1.5.16 pour Memcached

ElastiCache la version 1.5.16 pour Memcached ajoute le support pour la version 1.5.16 de Memcached. Sans aucune nouvelle fonctionnalité, il inclut des correctifs de bogue et des mises à jour cumulatives à partir des versions Memcached 1.5.14 et Memcached 1.5.15.

Pour plus d'informations, consultez les notes de mise à jour de Memcached 1.5.16 sur Memcached on. GitHub

ElastiCache version 1.5.10 pour Memcached

ElastiCache la version 1.5.10 pour Memcached prend en charge les fonctionnalités suivantes de Memcached :

- Rééquilibrage de section automatisé.
- Recherches plus rapides dans les tables de hachage avec l'algorithme `murmur3`.
- Algorithme LRU segmenté.
- Crawler LRU pour la récupération de mémoire d'arrière-plan
- `--enable-seccomp` : option pour la compilation.

Elle introduit aussi les paramètres `no_modern` et `inline_ascii_resp`. Pour de plus amples informations, veuillez consulter Modifications des paramètres Memcached 1.5.10.

Les améliorations apportées à Memcached depuis la ElastiCache version 1.4.34 pour Memcached sont les suivantes :

- Correctifs cumulés, par exemple opérations de lecture multiples ASCII, CVE-2017-9951 et analyses des limites pour `metadumper`.
- Meilleure gestion des connexions en fermant les connexions à la limite de connexion.
- Gestion améliorée de la taille d'élément pour une taille d'élément supérieure à 1 Mo.

- Meilleures performances et améliorations de la surcharge de mémoire en réduisant les besoins en mémoire par élément de quelques octets.

Pour plus d'informations, consultez les notes de mise à jour de Memcached 1.5.10 sur Memcached on. GitHub

ElastiCache version 1.4.34 pour Memcached

ElastiCache la version 1.4.34 pour Memcached n'ajoute aucune nouvelle fonctionnalité à la version 1.4.33. La version 1.4.34 est une version de correctifs qui est plus volumineuse que les versions habituelles de ce type.

Pour plus d'informations, consultez les notes de mise à jour de Memcached 1.4.34 sur Memcached on. GitHub

ElastiCache version 1.4.33 pour Memcached

Les améliorations ajoutées depuis la version 1.4.24 sont les suivantes :

- Possibilité de vider toutes les métadonnées pour une classe de sections particulière, une liste de classes de sections ou toutes les classes de sections. Pour plus d'informations, consultez Notes de mise à jour de Memcached 1.4.31.
- Amélioration de la prise en charge pour les éléments volumineux dépassant la taille par défaut de 1 mégaoctet. Pour plus d'informations, consultez Notes de mise à jour de Memcached 1.4.29.
- Possibilité de spécifier pendant combien de temps un client peut rester inactif avant d'être invité à se fermer.

  Possibilité d'augmenter dynamiquement la quantité de mémoire disponible pour Memcached sans avoir à redémarrer le cluster. Pour plus d'informations, consultez Notes de mise à jour de Memcached 1.4.27.
- La journalisation de `fetchers`, `mutations` et `evictions` est désormais prise en charge. Pour plus d'informations, consultez Notes de mise à jour de Memcached 1.4.26.
- La mémoire libérée peut être récupérée dans un pool global et réaffectée à de nouvelles classes de sections. Pour plus d'informations, consultez Notes de mise à jour de Memcached 1.4.25.
- Plusieurs corrections de bugs.
- Quelques nouvelles commandes et paramètres. Pour en obtenir la liste, veuillez consulter Paramètres ajoutés Memcached 1.4.33.

ElastiCache version 1.4.24 pour Memcached

Les améliorations ajoutées depuis la version 1.4.14 sont les suivantes :

- Une gestion de type LRU (Last Recently Used, moins récemment utilisé) à l'aide d'un processus d'arrière-plan.
- L'ajout de l'option d'utilisation de jenkins ou murmur3 en tant qu'algorithme de hachage.
- Quelques nouvelles commandes et paramètres. Pour en obtenir la liste, veuillez consulter Paramètres ajoutés Memcached 1.4.24.
- Plusieurs corrections de bugs.

ElastiCache version 1.4.14 pour Memcached

Les améliorations apportées depuis la version 1.4.5 sont les suivantes :

- Capacité de rééquilibrage de section améliorée.
- Amélioration de l'évolutivité et des performances.
- L'ajout de la commande touch pour mettre à jour le délai d'expiration d'un élément existant sans avoir à l'extraire.
- Découverte automatique : la possibilité pour les programmes client de déterminer automatiquement tous les nœuds de cache dans un cluster, et d'initier et maintenir des connexions à tous ces nœuds.

ElastiCache version 1.4.5 pour Memcached

ElastiCache la version 1.4.5 pour Memcached était le moteur initial et la version pris en charge par Amazon ElastiCache pour Memcached.

Principales différences de comportement entre les versions du moteur et de compatibilité avec Valkey

Valkey 7.2.6 présente des différences de compatibilité similaires avec les versions précédentes de Redis OSS 7.2.4. Pour la dernière version prise en charge de Valkey, consultezMoteurs et versions pris en charge.

Pour plus d'informations sur la version 7.2 de Valkey, consultez les notes de mise à jour de Redis OSS 7.2.4 (Valkey 7.2 inclut toutes les modifications depuis Redis OSS jusqu'à la version 7.2.4) et les notes de version de Valkey 7.2 sur Valkey on. GitHub

Voici les changements de comportement potentiellement perturbants entre Valkey 7.2 et Redis OSS 7.1 (ou 7.0) :

- L'échantillonnage pendant le temps de gel a lieu lors de l'exécution des commandes et dans les scripts.
- Une commande de diffusion bloquée lancée lorsque la clé n'existe plus contient un code d'erreur différent (-NOGROUP ou -WRONGTYPE au lieu de -UNBLOCKED).
- Le suivi côté client pour les scripts suit désormais les clés lues par le script, au lieu des clés déclarées par l'appelant de EVAL/FCALL.

Principales différences de comportement entre les versions du moteur et de compatibilité avec Redis OSS

> **Important**
>
> La page suivante est structurée de manière à indiquer toutes les différences d'incompatibilité entre les versions et à vous informer des éléments à prendre en compte lors de la mise à niveau vers des versions plus récentes. Cette liste inclut tous les problèmes d'incompatibilité de versions que vous pourriez rencontrer lors de la mise à niveau.
> Vous pouvez effectuer une mise à niveau directement de votre version actuelle de Redis OSS vers la dernière version de Redis OSS disponible, sans avoir besoin de mises à niveau séquentielles. Par exemple, vous pouvez passer directement de la version 3.0 de Redis OSS à la version 7.0.

Les versions Redis OSS sont identifiées par une version sémantique qui comprend un composant majeur, un composant mineur et un composant correctif. Par exemple, dans Redis OSS 4.0.10, la version principale est 4, la version mineure 0 et la version du correctif est 10. Ces valeurs sont généralement incrémentées en fonction des conventions suivantes :

- Les versions majeures concernent les modifications incompatibles avec l'API
- Les versions mineures concernent les nouvelles fonctionnalités ajoutées de manière rétrocompatible
- Les versions de correctif sont destinées aux corrections de bogues rétrocompatibles et aux modifications non fonctionnelles

Nous vous recommandons de toujours utiliser la dernière version du correctif au sein d'une version majeure .minor donnée afin de bénéficier des dernières améliorations de performances et de stabilité. À partir de ElastiCache la version 6.0 pour Redis OSS, nous ElastiCache proposerons une version unique pour chaque version mineure de Redis OSS plutôt que de proposer plusieurs versions de correctif. ElastiCachegérera automatiquement la version du correctif de vos clusters de cache en cours d'exécution, garantissant ainsi de meilleures performances et une sécurité renforcée.

Nous recommandons également de procéder périodiquement à une mise à niveau vers la dernière version majeure, car la plupart des améliorations majeures ne sont pas rétroportées vers des versions plus anciennes. À mesure que la disponibilité ElastiCache s'étend à une nouvelle AWS région, ElastiCache Redis OSS prend en charge les deux versions majeures .minor les plus récentes à l'époque pour la nouvelle région. Par exemple, si une nouvelle AWS région est lancée et que les dernières ElastiCache versions majeures de .minor pour Redis OSS sont 7.0 et 6.2, les versions 7.0 et 6.2 de Redis OSS ElastiCache seront prises en charge dans la nouvelle région. AWS Au fur et à mesure que les nouvelles versions majeures .minor de ElastiCache pour Redis OSS seront publiées, ElastiCache nous continuerons à ajouter le support pour les nouvelles versions. Pour en savoir plus sur le choix des régions pour ElastiCache, consultez la section Choix des régions et des zones de disponibilité.

Lorsque vous effectuez une mise à niveau qui couvre des versions majeures ou mineures, veuillez prendre en compte la liste suivante qui inclut les modifications comportementales et rétroincompatibles publiées avec Redis OSS au fil du temps.

Comportement de Redis OSS 7.0 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 7.0.

- `SCRIPT LOAD` et `SCRIPT FLUSH` ne sont plus propagés vers des réplicas. Si vous avez besoin d'une certaine durabilité pour les scripts, nous vous recommandons d'utiliser les fonctions Redis OSS.
- Les canaux Pubsub sont désormais bloqués par défaut pour les nouveaux utilisateurs de la liste ACL.
- La commande `STRALGO` a été remplacée par la commande `LCS`.
- Le format de `ACL GETUSER` a été modifié de sorte que tous les champs affichent le modèle de chaîne d'accès standard. Si vous avez utilisé l'automatisation pour `ACL GETUSER`, vous devez vérifier qu'il gère l'un des deux formats.
- Les catégories d'ACL pour `SELECT`, `WAIT`, `ROLE`, `LASTSAVE`, `READONLY`, `READWRITE` et `ASKING` ont changé.
- La commande `INFO` affiche désormais les statistiques des commandes par sous-commande plutôt que dans les commandes de conteneur de niveau supérieur.
- Les valeurs de retour des commandes `LPOP`, `RPOP`, `ZPOPMIN` et `ZPOPMAX` ont changé dans certains cas limites. Si vous utilisez ces commandes, vous devez consulter les notes de mise à jour et évaluer si vous êtes concerné.
- Les commandes `SORT` et `SORT_RO` nécessitent désormais un accès à l'intégralité de l'espace de clés pour pouvoir utiliser les arguments `GET` et `BY`.

Comportement de Redis OSS 6.2 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 6.2.

- Les drapeaux ACL des commandes `TIME`, `ECHO`, `ROLE` et `LASTSAVE` ont été modifiés. Cela peut entraîner le rejet de commandes qui étaient précédemment acceptées et vice versa.

> **Note**
>
> Aucune de ces commandes ne modifie ou ne donne accès aux données.

- Lors de la mise à niveau depuis Redis OSS 6.0, l'ordre des key/value paires renvoyées par une réponse cartographique à un script lua est modifié. Si vos scripts utilisent `redis.setresp()` ou renvoient une carte (nouveauté de Redis OSS 6.0), considérez les implications d'une interruption du script lors des mises à niveau.

Comportement de Redis OSS 6.0 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 6.0.

- Le nombre maximum de bases de données autorisées a été diminué de 1,2 million à 10 mille. La valeur par défaut est 16, et nous vous déconseillons d'utiliser des valeurs beaucoup plus grandes car nous avons constaté des problèmes de performances et de mémoire.
- Définissez `AutoMinorVersionUpgrade` le paramètre sur yes et ElastiCache gérera la mise à niveau de la version mineure via des mises à jour en libre-service. Cela sera géré par les canaux standard de notification client via une campagne de mise à jour en libre-service. Pour plus d'informations, consultez la section Mises à jour en libre-service dans ElastiCache.

Comportement de Redis OSS 5.0 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 5.0.

- Les scripts sont répliqués par effets au lieu de réexécuter le script sur le réplica. Cela améliore généralement les performances, mais peut augmenter la quantité de données répliquées entre les principaux et les réplicas. Il existe une option permettant de revenir au comportement précédent qui n'est disponible que dans la ElastiCache version 5.0 pour Redis OSS.
- Si vous effectuez une mise à niveau depuis Redis OSS 4.0, certaines commandes des scripts LUA renverront les arguments dans un ordre différent de celui des versions précédentes. Dans Redis OSS 4.0, Redis OSS ordonnait certaines réponses de manière lexographique afin de les rendre déterministes. Cet ordre n'est pas appliqué lorsque les scripts sont répliqués par des effets.
- Dans Redis OSS 5.0.3 et versions ultérieures, ElastiCache Redis OSS déchargera une partie du travail d'E/S vers les cœurs d'arrière-plan sur les types d'instances comportant plus de 4. VCPUs Cela peut modifier les caractéristiques de performance de Redis OSS et modifier les valeurs de certaines métriques. Pour plus d'informations, consultez Quelles métriques dois-je surveiller ? pour savoir si vous devez modifier les métriques que vous surveillez.

Comportement de Redis OSS 4.0 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 4.0.

- Le journal lent enregistre désormais deux arguments supplémentaires, le nom et l'adresse du client. Cette modification devrait être rétrocompatible, sauf si vous comptez explicitement sur le fait que chaque entrée du journal lent contient 3 valeurs.

- La commande `CLUSTER NODES` renvoie désormais un format légèrement différent, qui n'est pas rétrocompatible. Nous recommandons aux clients de ne pas utiliser cette commande pour connaître les nœuds présents dans un cluster, et d'utiliser plutôt `CLUSTER SLOTS`.

EOL passée

Comportement de Redis OSS 3.2 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 3.2.

- Il n'y a pas de modifications de compatibilité à signaler pour cette version.

Pour de plus amples informations, veuillez consulter ElastiCache versions pour le calendrier de fin de vie de Redis OSS.

Comportement de Redis OSS 2.8 et modifications rétroincompatibles

Pour une liste complète des modifications, consultez les notes de mise à jour de Redis OSS 2.8.

- À partir de Redis OSS 2.8.22, Redis OSS AOF n'est plus pris en charge dans Redis OSS. ElastiCache Nous recommandons d'utiliser MemoryDB lorsque les données doivent être conservées de manière durable.
- À partir de Redis OSS 2.8.22, ElastiCache Redis OSS ne prend plus en charge l'attachement de répliques aux serveurs principaux hébergés dans Redis OSS. ElastiCache Pendant la mise à niveau, les réplicas externes seront déconnectés et il sera impossible de les reconnecter. Nous recommandons d'utiliser la mise en cache côté client, disponible dans Redis OSS 6.0, comme alternative aux répliques externes.
- Les commandes `TTL` et `PTTL` renvoient désormais -2 si la clé n'existe pas et -1 si elle existe, mais n'a pas d'expiration associée. Redis OSS 2.6 et les versions précédentes renvoyaient -1 pour les deux conditions.
- `SORT` avec `ALPHA` trie désormais en fonction des paramètres régionaux de classement locaux si aucune option `STORE` n'est utilisée.

Pour de plus amples informations, veuillez consulter ElastiCache versions pour le calendrier de fin de vie de Redis OSS.

Considérations en matière de mise à niveau lorsque vous utilisez des clusters auto-conçus

> **Note**
>
> Les considérations suivantes s'appliquent uniquement lors de la mise à niveau de clusters auto-conçus. Ils ne s'appliquent pas à ElastiCache Serverless.

Considérations relatives à Valkey et Redis OSS

Lors de la mise à niveau d'un cluster Valkey ou Redis OSS conçu par vos soins, tenez compte des points suivants.

- La gestion de la version du moteur est conçue afin que vous ayez autant de contrôle que possible sur le déroulement de la correction. Toutefois, ElastiCache se réserve 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 de cache.
- À partir de ElastiCache la version 7.2 pour Valkey et de ElastiCache la version 6.0 pour Redis OSS, nous ElastiCache proposerons une version unique pour chaque version mineure, plutôt que de proposer plusieurs versions de correctif.
- À partir de la version 5.0.6 du moteur Redis OSS, 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.
- Vous pouvez également mettre à niveau vos ElastiCache clusters avec des versions antérieures à 5.0.6. Le processus impliqué est le même, mais peut entraîner un temps de basculement plus long pendant la propagation DNS (30 s-1 mn).
- À partir de Redis OSS 7, ElastiCache permet de basculer entre Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé).
- Le processus de mise à niveau du moteur Amazon ElastiCache for Redis OSS est conçu pour faire de son mieux pour conserver vos données existantes et nécessite une réplication Redis OSS réussie.
- Lors de la mise à niveau du moteur, les connexions client existantes ElastiCache seront interrompues. Pour minimiser les temps d'arrêt lors des mises à niveau du moteur, nous vous

recommandons de mettre en œuvre les meilleures pratiques pour les clients Redis OSS, avec des tentatives d'erreur et des retards exponentiels, ainsi que les meilleures pratiques pour minimiser les temps d'arrêt pendant la maintenance.

- Vous ne pouvez pas passer directement de Valkey ou Redis OSS (mode cluster désactivé) vers Valkey ou Redis OSS (mode cluster activé) lorsque vous mettez à niveau votre moteur. La procédure suivante explique comment passer de Valkey ou Redis OSS (mode cluster désactivé) vers Valkey ou Redis OSS (mode cluster activé).

  Pour passer d'une version de moteur Valkey ou Redis OSS (mode cluster désactivé) vers une version de moteur Valkey ou Redis OSS (mode cluster activé)

  1. Effectuez une sauvegarde de votre cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour de plus amples informations, veuillez consulter Réalisation de sauvegardes manuelles.
  2. Utilisez la sauvegarde pour créer et amorcer un cluster Valkey ou Redis OSS (mode cluster activé) avec une partition (groupe de nœuds). Spécifiez la nouvelle version du moteur et activez le mode de cluster lors de la création du cluster ou du groupe de réplication. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.
  3. Supprimez l'ancien cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour plus d'informations, consultez Supprimer un cluster dans ElastiCache ou Suppression d'un groupe de réplication.
  4. Adaptez le nouveau cluster ou groupe de réplication Valkey ou Redis OSS (mode cluster activé) au nombre de partitions (groupes de nœuds) dont vous avez besoin. Pour de plus amples informations, consultez Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).
- Lors de la mise à niveau des versions majeures du moteur, par exemple de 5.0.6 à 6.0, vous devez également choisir un nouveau groupe de paramètres compatible avec la nouvelle version du moteur.
- Pour les clusters Redis OSS uniques et les clusters dont le mode multi-AZ est désactivé, nous recommandons de mettre suffisamment de mémoire à la disposition de Redis OSS, comme décrit dans. S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS Dans ce cas, le réplica principal n'est pas disponible pour traiter les demandes de service pendant la mise à niveau.
- Pour les clusters Redis OSS sur lesquels le mode multi-AZ est activé, nous vous recommandons également de planifier les mises à niveau du moteur pendant les périodes de faible trafic d'écriture

entrant. Lors de la mise à niveau vers Redis OSS 5.0.6 ou une version ultérieure, le cluster principal reste disponible pour les demandes de service pendant le processus de mise à niveau.

Les clusters et les groupes de réplication avec plusieurs partitions sont traités et soumis à des correctifs comme suit :

- Toutes les partitions sont traitées en parallèle. Une seule opération de mise à niveau à la fois est effectuée sur une partition.
- 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.

- Si les chiffrements sont activés sur votre cluster ou votre groupe de réplication actuel, vous ne pouvez pas effectuer de mise à niveau vers une version du moteur ne prenant pas en charge le chiffrement, comme par exemple de 3.2.6 vers 3.2.10.

Considérations relatives à Memcached

Lorsque vous mettez à niveau un cluster Memcached conçu par vos soins, tenez compte des points suivants.

- La gestion de la version du moteur est conçue afin que vous ayez autant de contrôle que possible sur le déroulement de la correction. Toutefois, ElastiCache se réserve 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 de cache.
- Comme le moteur Memcached ne prend pas en charge la persistance, les mises à niveau de version du moteur Memcached sont toujours un processus perturbateur qui efface toutes les données de cache dans le cluster.

ElastiCache meilleures pratiques et stratégies de mise en cache

Vous trouverez ci-dessous les meilleures pratiques recommandées pour Amazon ElastiCache. La mise en œuvre de ces bonnes pratiques améliore les performances et la fiabilité de votre cache.

Rubriques

- Bonnes pratiques générales

- Bonnes pratiques d'utilisation des répliques de lecture
- Commandes Valkey, Memcached et Redis OSS prises en charge et restreintes
- Configuration et limites de Valkey et Redis OSS
- IPv6 exemples de clients pour Valkey, Memcached et Redis OSS
- Bonnes pratiques pour les clients (Valkey et Redis OSS)
- Bonnes pratiques pour les clients (Memcached)
- Clusters à double pile ElastiCache compatibles TLS
- Gestion de la mémoire réservée pour Valkey et Redis OSS
- Meilleures pratiques lors de l'utilisation de clusters conçus par Valkey et Redis OSS
- Stratégies de mise en cache pour Memcached

Bonnes pratiques générales

Vous trouverez ci-dessous des informations sur les meilleures pratiques d'utilisation des interfaces Valkey, Memcached et Redis OSS. ElastiCache

- Utiliser des configurations activées en mode cluster : le mode cluster activé permet au cache de s'adapter horizontalement pour obtenir un stockage et un débit supérieurs à ceux d'une configuration désactivée en mode cluster. ElastiCache serverless n'est disponible que dans une configuration activée en mode cluster.
- Utilisez des connexions longue durée : la création d'une connexion est coûteuse et nécessite du temps et des ressources de CPU provenant du cache. Réutilisez les connexions dans la mesure du possible (en regroupant les connexions, par exemple) pour amortir ce coût sur de nombreuses commandes.
- Lecture à partir de répliques : si vous utilisez des répliques ElastiCache sans serveur ou si vous avez provisionné des répliques en lecture (clusters conçus par vos soins), effectuez des lectures directes vers des répliques pour obtenir une meilleure évolutivité et réduire la latence. and/or Les lectures de réplicas sont cohérentes avec le nœud primaire à terme.

  Dans un cluster auto-conçu, évitez de diriger les demandes de lecture vers un seul réplica en lecture, car les lectures risquent de ne pas être disponibles temporairement en cas de défaillance du nœud. Configurez votre client pour qu'il dirige les demandes de lecture vers au moins deux réplicas en lecture, ou qu'il dirige les lectures vers un seul réplica et le nœud primaire.

En mode ElastiCache sans serveur, la lecture depuis le port de réplication (6380) dirige les lectures vers la zone de disponibilité locale du client lorsque cela est possible, réduisant ainsi la latence de récupération. Il retombera automatiquement sur les autres nœuds en cas de défaillance.

- Évitez les commandes coûteuses : évitez d'exécuter des opérations de calcul I/O intensives, 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.
- Utiliser un pub/sub fragmenté : lorsque vous utilisez Valkey ou Redis OSS pour prendre en charge des pub/sub charges de travail à haut débit, nous vous recommandons d'utiliser un pub/sub fragmenté (disponible avec Valkey et avec Redis OSS 7 ou version ultérieure). Les clusters traditionnels activés pub/sub en mode cluster diffusent des messages à tous les nœuds du cluster, ce qui peut entraîner une hausse. `EngineCPUUtilization` Notez que dans les pub/sub commands internally use sharded pub/sub commandes traditionnelles ElastiCache sans serveur.

Bonnes pratiques d'utilisation des répliques de lecture

De nombreuses applications, telles que les magasins de sessions, les classements et les moteurs de recommandation, nécessitent une haute disponibilité et gèrent beaucoup plus d'opérations de lecture que d'opérations d'écriture. Ces applications peuvent souvent tolérer des données légèrement périmées (cohérence éventuelle), ce qui signifie qu'il est acceptable que différents utilisateurs voient momentanément des versions légèrement différentes des mêmes données. Par exemple :

- Les résultats des requêtes mis en cache peuvent souvent tolérer des données légèrement périmées, en particulier pour les modèles mis en cache où la source de vérité est externe.
- Dans un classement de jeu, un retard de quelques secondes dans la mise à jour des scores n'a souvent pas d'impact significatif sur l'expérience utilisateur.
- En ce qui concerne les stockages de sessions, certains légers retards dans la propagation des données de session entre les répliques affectent rarement les fonctionnalités de l'application.
- Les moteurs de recommandation utilisent généralement l'analyse des données historiques, de sorte que la cohérence en temps réel est moins critique.

La cohérence éventuelle signifie que tous les nœuds de réplication renverront les mêmes données une fois le processus de réplication terminé, généralement en quelques millisecondes. Dans de tels cas d'utilisation, la mise en œuvre de répliques de lecture constitue une stratégie efficace pour réduire le temps de latence lors de la lecture depuis votre ElastiCache instance.

L'utilisation de répliques de lecture dans Amazon ElastiCache peut apporter des avantages significatifs en termes de performances grâce à :

Évolutivité de lecture améliorée

- Répartit les opérations de lecture sur plusieurs nœuds de réplication
- Décharge le trafic de lecture du nœud principal
- Réduit la latence de lecture en répondant aux demandes provenant de répliques géographiquement plus proches

Performances optimisées du nœud principal

- Dédie les ressources du nœud principal aux opérations d'écriture
- Réduit les frais de connexion sur le nœud principal
- Améliore les performances d'écriture et maintient de meilleurs temps de réponse pendant les périodes de pointe

Utilisation de Read from Replica en mode ElastiCache Serverless

ElastiCache serverless fournit deux points de terminaison différents, pour des exigences de cohérence différentes. Les deux points de terminaison utilisent le même nom DNS mais des ports différents. Pour utiliser le read-from-replica port, vous devez autoriser l'accès aux deux ports depuis votre application cliente en configurant les groupes de sécurité et les listes de contrôle d'accès réseau de votre VPC.

Point de terminaison principal (port 6379)

- Utilisation pour les opérations nécessitant une cohérence immédiate
- Garantit la lecture du plus grand nombre up-to-date de données
- Idéal pour les transactions critiques et les opérations d'écriture
- Nécessaire pour les opérations d'écriture
- Exemple : `test-12345.serverless.use1.cache.amazonaws.com:6379`

Point de terminaison optimisé pour la latence (port 6380)

- Optimisé pour les opérations de lecture qui peuvent tolérer une éventuelle cohérence
- Lorsque cela est possible, le mode ElastiCache sans serveur achemine automatiquement les demandes de lecture vers le nœud de réplication dans la zone de disponibilité locale du client. Cette optimisation permet de réduire la latence en évitant la latence réseau supplémentaire encourue lors de la récupération de données à partir d'un nœud situé dans une autre zone de disponibilité.
- ElastiCache serverless sélectionne automatiquement les nœuds disponibles dans d'autres zones si un nœud local n'est pas disponible
- Exemple : `test-12345.serverless.use1.cache.amazonaws.com:6380`
- Les clients tels que Glide et Lettuce détectent et acheminent automatiquement les lectures vers le point de terminaison optimisé pour la latence si vous fournissez la configuration de lecture depuis une réplique. Si votre client ne prend pas en charge la configuration du routage (par exemple, Valkey-Java et les anciennes versions de Jedis), vous devez définir le port et la configuration client appropriés pour lire les répliques.

Connexion pour lire des répliques en mode ElastiCache Serverless - Valkey and Glide

L'extrait de code suivant montre comment configurer la lecture depuis une réplique pour ElastiCache Serverless dans la bibliothèque Valkey Glide. Il n'est pas nécessaire de spécifier le port pour la lecture à partir des répliques, mais vous devez configurer la configuration `ReadFrom.PREFER_REPLICA` du routage.

```
package glide.examples;

import glide.api.GlideClusterClient;
import glide.api.logging.Logger;
import glide.api.models.configuration.GlideClusterClientConfiguration;
import glide.api.models.configuration.NodeAddress;
import glide.api.models.exceptions.ClosingException;
import glide.api.models.exceptions.ConnectionException;
import glide.api.models.exceptions.TimeoutException;
import glide.api.models.configuration.ReadFrom;

import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutionException;

public class ClusterExample {
```

```
    public static void main(String[] args) {
        // Set logger configuration
        Logger.setLoggerConfig(Logger.Level.INFO);

        GlideClusterClient client = null;

        try {
            System.out.println("Connecting to Valkey Glide...");

            // Configure the Glide Client
            GlideClusterClientConfiguration config =
GlideClusterClientConfiguration.builder()
                .address(NodeAddress.builder()
                    .host("your-endpoint")
                    .port(6379)
                    .build())
                .useTLS(true)
                .readFrom(ReadFrom.PREFER_REPLICA)
                .build();

            // Create the GlideClusterClient
            client = GlideClusterClient.createClient(config).get();
            System.out.println("Connected successfully.");

            // Perform SET operation
            CompletableFuture<String> setResponse = client.set("key", "value");
            System.out.println("Set key 'key' to 'value': " + setResponse.get());

            // Perform GET operation
            CompletableFuture<String> getResponse = client.get("key");
            System.out.println("Get response for 'key': " + getResponse.get());

            // Perform PING operation
            CompletableFuture<String> pingResponse = client.ping();
            System.out.println("PING response: " + pingResponse.get());

        } catch (ClosingException | ConnectionException | TimeoutException |
ExecutionException e) {
            System.err.println("An exception occurred: ");
            e.printStackTrace();
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        } finally {
```

```
            // Close the client connection
            if (client != null) {
                try {
                    client.close();
                    System.out.println("Client connection closed.");
                } catch (ClosingException | ExecutionException e) {
                    System.err.println("Error closing client: " + e.getMessage());
                }
            }
        }
    }
}
```

Commandes Valkey, Memcached et Redis OSS prises en charge et restreintes

Commandes Valkey et Redis OSS prises en charge

Commandes Valkey et Redis OSS prises en charge

Les commandes Valkey et Redis OSS suivantes sont prises en charge par les caches sans serveur. Outre ces commandes, les commandes répertoriées à la rubrique Commandes Valkey et Redis OSS prises en charge sont également prises en charge.

Pour plus d'informations sur les commandes du filtre Bloom, voir Commandes du filtre Bloom

Commandes Bitmap

- `BITCOUNT`

  Compte le nombre de bits définis (comptage de la population) dans une chaîne.

  En savoir plus
- `BITFIELD`

  Effectue des opérations arbitraires sur des champs de bits entiers dans des chaînes.

  En savoir plus
- `BITFIELD_RO`

  Effectue des opérations arbitraires sur des champs de bits entiers en lecture seule dans des chaînes.

En savoir plus

- `BITOP`

  Effectue des opérations au niveau du bit sur plusieurs chaînes et stocke le résultat.

  En savoir plus

- `BITPOS`

  Recherche le premier bit défini (1) ou effacé (0) dans une chaîne.

  En savoir plus

- `GETBIT`

  Renvoie une valeur de bit par décalage.

  En savoir plus

- `SETBIT`

  Définit ou efface le bit décalé par rapport à la valeur de chaîne. Crée la clé si elle n'existe pas.

  En savoir plus

Commandes de gestion des clusters

- `CLUSTER COUNTKEYSINSLOT`

  Renvoie le nombre de clés dans un emplacement de hachage.

  En savoir plus

- `CLUSTER GETKEYSINSLOT`

  Renvoie les noms de clés dans un emplacement de hachage.

  En savoir plus

- `CLUSTER INFO`

  Renvoie des informations sur l'état d'un nœud. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

En savoir plus

- `CLUSTER KEYSLOT`

  Renvoie l'emplacement de hachage d'une clé.

  En savoir plus

- `CLUSTER MYID`

  Renvoie l'ID d'un nœud. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

  En savoir plus

- `CLUSTER NODES`

  Renvoie la configuration de cluster d'un nœud. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

  En savoir plus

- `CLUSTER REPLICAS`

  Répertorie les nœuds de réplica d'un nœud principal. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

  En savoir plus

- `CLUSTER SHARDS`

  Renvoie le mappage des emplacements de cluster aux partitions. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

  En savoir plus

- `CLUSTER SLOTS`

  Renvoie le mappage des emplacements de cluster aux nœuds. Dans un cache sans serveur, renvoie l'état de l'unique « partition » virtuelle exposée au client.

  En savoir plus

- `CLUSTER SLOT-STATS`

Permet le suivi des métriques par slot pour le nombre de clés, l'utilisation du processeur, les octets réseau entrants et les octets réseau sortants.

En savoir plus

- `READONLY`

  Active les requêtes en lecture seule pour une connexion à un nœud de réplication Valkey ou Redis OSS Cluster.

  En savoir plus

- `READWRITE`

  Active les requêtes en lecture-écriture pour une connexion à un nœud de réplication Valkey ou Redis OSS Cluster.

  En savoir plus

- `SCRIPT SHOW`

  Renvoie le code source d'origine d'un script dans le cache de scripts.

  En savoir plus

Commandes de gestion des connexions

- `AUTH`

  Authentifie la connexion.

  En savoir plus

- `CLIENT GETNAME`

  Renvoie le nom de la connexion.

  En savoir plus

- `CLIENT REPLY`

  Demande au serveur de répondre ou non aux commandes.

  En savoir plus

- `CLIENT SETNAME`

  Définit le nom de la connexion.

  En savoir plus

- `ECHO`

  Renvoie la chaîne donnée.

  En savoir plus

- `HELLO`

  Poignées de main avec le serveur Valkey ou Redis OSS.

  En savoir plus

- `PING`

  Indique si la connexion au serveur est toujours active.

  En savoir plus

- `QUIT`

  Ferme la connexion.

  En savoir plus

- `RESET`

  Réinitialise la connexion.

  En savoir plus

- `SELECT`

  Modifie la base de données sélectionnée.

  En savoir plus

Commandes génériques

- `COPY`

Copie la valeur d'une clé dans une nouvelle clé.

En savoir plus

- `DEL`

  Supprime une ou plusieurs clés.

  En savoir plus

- `DUMP`

  Renvoie une représentation sérialisée de la valeur stockée dans une clé.

  En savoir plus

- `EXISTS`

  Détermine si une ou plusieurs clés existent.

  En savoir plus

- `EXPIRE`

  Définit le délai d'expiration d'une clé en secondes.

  En savoir plus

- `EXPIREAT`

  Définit le délai d'expiration d'une clé selon un horodatage Unix.

  En savoir plus

- `EXPIRETIME`

  Renvoie le délai d'expiration d'une clé au format d'horodatage Unix.

  En savoir plus

- `PERSIST`

  Supprime le délai d'expiration d'une clé.

  En savoir plus

- `PEXPIRE`

Définit le délai d'expiration d'une clé en millisecondes.

En savoir plus

- `PEXPIREAT`

  Définit le délai d'expiration d'une clé selon un horodatage Unix en millisecondes.

  En savoir plus

- `PEXPIRETIME`

  Renvoie le délai d'expiration d'une clé au format d'horodatage Unix en millisecondes.

  En savoir plus

- `PTTL`

  Renvoie le délai d'expiration d'une clé en millisecondes.

  En savoir plus

- `RANDOMKEY`

  Renvoie un nom de clé aléatoire à partir de la base de données.

  En savoir plus

- `RENAME`

  Renomme une clé et remplace la destination.

  En savoir plus

- `RENAMENX`

  Renomme une clé uniquement lorsque le nom de la clé cible n'existe pas.

  En savoir plus

- `RESTORE`

  Crée une clé à partir de la représentation sérialisée d'une valeur.

  En savoir plus

- `SCAN`

Effectue une itération sur les noms de clés de la base de données.

En savoir plus

- `SORT`

  Trie les éléments d'une liste, d'un ensemble ou d'un ensemble trié, en stockant éventuellement le résultat.

  En savoir plus

- `SORT_RO`

  Renvoie les éléments triés d'une liste, d'un ensemble ou d'un ensemble trié.

  En savoir plus

- `TOUCH`

  Renvoie le nombre de clés existantes parmi celles spécifiées après la mise à jour de l'heure de leur dernier accès.

  En savoir plus

- `TTL`

  Renvoie le délai d'expiration d'une clé en secondes.

  En savoir plus

- `TYPE`

  Détermine le type de valeur stockée dans une clé.

  En savoir plus

- `UNLINK`

  Supprime une ou plusieurs clés de manière asynchrone.

  En savoir plus

Commandes géospatiales

- `GEOADD`

Ajoute un ou plusieurs membres à un index géospatial. La clé est créée si elle n'existe pas.

En savoir plus

- `GEODIST`

  Renvoie la distance entre deux membres d'un index géospatial.

  En savoir plus

- `GEOHASH`

  Renvoie les membres d'un index géospatial sous forme de chaînes geohash.

  En savoir plus

- `GEOPOS`

  Renvoie la longitude et la latitude des membres à partir d'un index géospatial.

  En savoir plus

- `GEORADIUS`

  Recherche dans un index géospatial les membres situés à une certaine distance d'une coordonnée, puis stocke éventuellement le résultat.

  En savoir plus

- `GEORADIUS_RO`

  Renvoie les membres d'un index géospatial situés à une certaine distance d'une coordonnée.

  En savoir plus

- `GEORADIUSBYMEMBER`

  Recherche dans un index géospatial les membres situés à une certaine distance d'un membre, puis stocke éventuellement le résultat.

  En savoir plus

- `GEORADIUSBYMEMBER_RO`

  Renvoie les membres d'un index géospatial situés à une certaine distance d'un membre.

En savoir plus

- `GEOSEARCH`

  Recherche dans un index géospatial les membres situés à l'intérieur une zone d'une boîte ou d'un cercle.

  En savoir plus

- `GEOSEARCHSTORE`

  Recherche dans un index géospatial les membres situés à l'intérieur une zone d'une boîte ou d'un cercle, puis stocke éventuellement le résultat.

  En savoir plus

Commandes de hachage

- `HDEL`

  Supprime un ou plusieurs champs et leurs valeurs d'un hachage. Supprime le hachage s'il ne reste aucun champ.

  En savoir plus

- `HEXISTS`

  Détermine si un champ existe dans un hachage.

  En savoir plus

- `HGET`

  Renvoie la valeur d'un champ dans un hachage.

  En savoir plus

- `HGETALL`

  Renvoie tous les champs et toutes les valeurs d'un hachage.

  En savoir plus

- `HINCRBY`

Incrémente d'un nombre la valeur entière d'un champ dans un hachage. Utilise 0 comme valeur initiale si le champ n'existe pas.

En savoir plus

- `HINCRBYFLOAT`

  Incrémente d'un nombre la valeur à virgule flottante d'un champ. Utilise 0 comme valeur initiale si le champ n'existe pas.

  En savoir plus

- `HKEYS`

  Renvoie tous les champs d'un hachage.

  En savoir plus

- `HLEN`

  Renvoie le nombre de champs d'un hachage.

  En savoir plus

- `HMGET`

  Renvoie les valeurs de tous les champs d'un hachage.

  En savoir plus

- `HMSET`

  Définit les valeurs de plusieurs champs.

  En savoir plus

- `HRANDFIELD`

  Renvoie un ou plusieurs champs aléatoires à partir d'un hachage.

  En savoir plus

- `HSCAN`

  Effectue une itération sur les champs et les valeurs d'un hachage.

En savoir plus

- `HSET`

  Crée ou modifie la valeur d'un champ dans un hachage.

  En savoir plus

- `HSETNX`

  Définit la valeur d'un champ dans un hachage uniquement lorsque le champ n'existe pas.

  En savoir plus

- `HSTRLEN`

  Renvoie la longueur de la valeur d'un champ.

  En savoir plus

- `HVALS`

  Renvoie toutes les valeurs d'un hachage.

  En savoir plus

HyperLogLog Commandes

- `PFADD`

  Ajoute des éléments à une HyperLogLog clé. Crée la clé si elle n'existe pas.

  En savoir plus

- `PFCOUNT`

  Renvoie la cardinalité approximative du ou des ensembles observés par la ou les HyperLogLog touches.

  En savoir plus

- `PFMERGE`

  Fusionne une ou plusieurs HyperLogLog valeurs en une seule clé.

En savoir plus

Commandes de liste

- `BLMOVE`

  Extrait un élément d'une liste, le transmet à une autre liste et le renvoie. Bloque jusqu'à ce qu'un élément soit disponible autrement. Supprime la liste si le dernier élément a été déplacé.

  En savoir plus
- `BLMPOP`

  Extrait le premier élément de l'une des listes. Bloque jusqu'à ce qu'un élément soit disponible autrement. Supprime la liste si le dernier élément a été extrait.

  En savoir plus
- `BLPOP`

  Supprime et renvoie le premier élément d'une liste. Bloque jusqu'à ce qu'un élément soit disponible autrement. Supprime la liste si le dernier élément a été extrait.

  En savoir plus
- `BRPOP`

  Supprime et renvoie le dernier élément d'une liste. Bloque jusqu'à ce qu'un élément soit disponible autrement. Supprime la liste si le dernier élément a été extrait.

  En savoir plus
- `BRPOPLPUSH`

  Extrait un élément d'une liste, le transmet à une autre liste et le renvoie. Bloque jusqu'à ce qu'un élément soit disponible autrement. Supprime la liste si le dernier élément a été extrait.

  En savoir plus
- `LINDEX`

  Renvoie un élément d'une liste en fonction de son index.

  En savoir plus

- `LINSERT`

  Insère un élément avant ou après un autre élément dans une liste.

  En savoir plus

- `LLEN`

  Renvoie la longueur d'une liste.

  En savoir plus

- `LMOVE`

  Renvoie un élément après l'avoir extrait d'une liste et l'avoir transféré vers une autre. Supprime la liste si le dernier élément a été déplacé.

  En savoir plus

- `LMPOP`

  Renvoie plusieurs éléments d'une liste après les avoir retirés. Supprime la liste si le dernier élément a été extrait.

  En savoir plus

- `LPOP`

  Renvoie les premiers éléments d'une liste après les avoir retirés. Supprime la liste si le dernier élément a été extrait.

  En savoir plus

- `LPOS`

  Renvoie l'index des éléments correspondants dans une liste.

  En savoir plus

- `LPUSH`

  Ajoute un ou plusieurs éléments à une liste. Crée la clé si elle n'existe pas.

  En savoir plus

- `LPUSHX`

Ajoute un ou plusieurs éléments avant une liste uniquement lorsque celle-ci existe.

En savoir plus

- `LRANGE`

  Renvoie un ensemble d'éléments à partir d'une liste.

  En savoir plus

- `LREM`

  Retire les éléments d'une liste. Supprime la liste si le dernier élément a été retiré.

  En savoir plus

- `LSET`

  Définit la valeur d'un élément d'une liste en fonction de son index.

  En savoir plus

- `LTRIM`

  Retire des éléments aux deux extrémités d'une liste. Supprime la liste si tous les éléments ont été tronqués.

  En savoir plus

- `RPOP`

  Renvoie et retire les derniers éléments d'une liste. Supprime la liste si le dernier élément a été extrait.

  En savoir plus

- `RPOPLPUSH`

  Renvoie le dernier élément d'une liste après l'avoir retiré et transféré vers une autre liste. Supprime la liste si le dernier élément a été extrait.

  En savoir plus

- `RPUSH`

  Ajoute un ou plusieurs éléments à une liste. Crée la clé si elle n'existe pas.

En savoir plus

- `RPUSHX`

  Ajoute un élément à une liste uniquement lorsque celle-ci existe.

  En savoir plus

Commandes Pub/Sub

> **Note**
> Étant donné que les commandes PUBSUB utilisent en interne un mécanisme PUBSUB partitionné, les noms de canaux seront mélangés.

- `PUBLISH`

  Publie un message sur un canal.

  En savoir plus

- `PUBSUB CHANNELS`

  Renvoie les canaux actifs.

  En savoir plus

- `PUBSUB NUMSUB`

  Renvoie le nombre d'abonnés aux canaux.

  En savoir plus

- `PUBSUB SHARDCHANNELS`

  Renvoie les canaux de partition actifs.

  En savoir plus

- `PUBSUB SHARDNUMSUB`

  Renvoie le nombre d'abonnés aux canaux de partition.

En savoir plus

- `SPUBLISH`

  Publie un message sur un canal de partition.

  En savoir plus

- `SSUBSCRIBE`

  Écoute les messages publiés sur les canaux de partition.

  En savoir plus

- `SUBSCRIBE`

  Écoute les messages publiés sur les canaux.

  En savoir plus

- `SUNSUBSCRIBE`

  Arrête d'écouter les messages publiés sur les canaux de partition.

  En savoir plus

- `UNSUBSCRIBE`

  Arrête d'écouter les messages publiés sur les canaux.

  En savoir plus

Commandes de création de script

- `EVAL`

  Exécute un script Lua côté serveur.

  En savoir plus

- `EVAL_RO`

  Exécute un script Lua côté serveur en lecture seule.

  En savoir plus

- `EVALSHA`

  Exécute un script Lua côté serveur par digest. SHA1

  En savoir plus
- `EVALSHA_RO`

  Exécute un script Lua en lecture seule côté serveur par digest. SHA1

  En savoir plus
- `SCRIPT EXISTS`

  Détermine si des scripts Lua côté serveur existent dans le cache de script.

  En savoir plus
- `SCRIPT FLUSH`

  Actuellement, un cache de script no-op est géré par le service.

  En savoir plus
- `SCRIPT LOAD`

  Charge un script Lua côté serveur dans le cache de script.

  En savoir plus

Commandes de gestion des serveurs

> **Note**
>
> Lorsque vous utilisez des ElastiCache clusters conçus par vos soins pour Valkey et Redis OSS, le client doit envoyer des commandes de vidage à chaque serveur principal pour vider toutes les clés. ElastiCache Serverless pour Valkey et Redis OSS fonctionne différemment, car il fait abstraction de la topologie de cluster sous-jacente. Le résultat est que dans ElastiCache Serverless, `FLUSHDB` les `FLUSHALL` commandes effleurent toujours toutes les clés du cluster. Pour cette raison, les commandes flush ne peuvent pas être incluses dans une transaction sans serveur.

- `ACL CAT`

  Répertorie les catégories de liste ACL ou les commandes d'une catégorie.

  En savoir plus
- `ACL GENPASS`

  Génère un mot de passe sécurisé pseudo-aléatoire qui peut être utilisé pour identifier les utilisateurs de liste ACL.

  En savoir plus
- `ACL GETUSER`

  Répertorie les règles de liste ACL d'un utilisateur.

  En savoir plus
- `ACL LIST`

  Vide les règles en vigueur au format de fichier ACL.

  En savoir plus
- `ACL USERS`

  Répertorie tous les utilisateurs de liste ACL.

  En savoir plus
- `ACL WHOAMI`

  Renvoie le nom d'utilisateur authentifié de la connexion en cours.

  En savoir plus
- `DBSIZE`

  Renvoie le nombre de clés dans la base de données actuellement sélectionnée. Il n'est pas garanti que cette opération soit atomique sur tous les emplacements.

  En savoir plus
- `COMMAND`

  Renvoie des informations détaillées sur toutes les commandes.

En savoir plus

- `COMMAND COUNT`

  Renvoie le nombre de commandes.

  En savoir plus

- `COMMAND DOCS`

  Renvoie des informations documentaires sur une, plusieurs ou toutes les commandes.

  En savoir plus

- `COMMAND GETKEYS`

  Extrait les noms de clés d'une commande arbitraire.

  En savoir plus

- `COMMAND GETKEYSANDFLAGS`

  Extrait les noms de clés et les indicateurs d'accès d'une commande arbitraire.

  En savoir plus

- `COMMAND INFO`

  Renvoie des informations sur une, plusieurs ou toutes les commandes.

  En savoir plus

- `COMMAND LIST`

  Renvoie la liste des noms de commandes.

  En savoir plus

- `COMMANDLOG`

  Un conteneur pour les commandes du journal des commandes.

  En savoir plus

- `COMMANDLOG GET`

  Renvoie les entrées du journal de commandes spécifié.

En savoir plus

- `COMMANDLOG HELP`

  Afficher un texte utile sur les différentes sous-commandes.

  En savoir plus

- `COMMANDLOG LEN`

  Renvoie le nombre d'entrées dans le type de journal de commandes spécifié.

  En savoir plus

- `COMMANDLOG RESET`

  Efface toutes les entrées du type de journal de commandes spécifié.

  En savoir plus

- `FLUSHALL`

  Retire toutes les clés de toutes les bases de données. Il n'est pas garanti que cette opération soit atomique sur tous les emplacements.

  En savoir plus

- `FLUSHDB`

  Retire toutes les clés de la base de données en cours. Il n'est pas garanti que cette opération soit atomique sur tous les emplacements.

  En savoir plus

- `INFO`

  Renvoie des informations et des statistiques sur le serveur.

  En savoir plus

- `LOLWUT`

  Affiche les illustrations informatiques et la version Valkey ou Redis OSS.

  En savoir plus

- `ROLE`

Renvoie le rôle de réplication.

En savoir plus

- `TIME`

  Renvoie l'heure du serveur.

  En savoir plus

Commandes d'ensemble

- `SADD`

  Ajoute un ou plusieurs membres à un ensemble. Crée la clé si elle n'existe pas.

  En savoir plus

- `SCARD`

  Renvoie le nombre de membres d'un ensemble.

  En savoir plus

- `SDIFF`

  Renvoie la différence entre plusieurs ensembles.

  En savoir plus

- `SDIFFSTORE`

  Stocke dans une clé la différence entre plusieurs ensembles.

  En savoir plus

- `SINTER`

  Renvoie l'intersection de plusieurs ensembles.

  En savoir plus

- `SINTERCARD`

  Renvoie le nombre de membres de l'intersection de plusieurs ensembles.

En savoir plus

- `SINTERSTORE`

  Stocke dans une clé l'intersection de plusieurs ensembles.

  En savoir plus

- `SISMEMBER`

  Détermine si un membre appartient à un ensemble.

  En savoir plus

- `SMEMBERS`

  Renvoie tous les membres d'un ensemble.

  En savoir plus

- `SMISMEMBER`

  Détermine si plusieurs membres appartiennent à un ensemble.

  En savoir plus

- `SMOVE`

  Déplace un membre d'un ensemble à un autre.

  En savoir plus

- `SPOP`

  Renvoie un ou plusieurs membres aléatoires d'un ensemble après les avoir retirés. Supprime l'ensemble si le dernier membre a été extrait.

  En savoir plus

- `SRANDMEMBER`

  Obtient un ou plusieurs membres aléatoires à partir d'un ensemble.

  En savoir plus

- `SREM`

Retire un ou plusieurs membres d'un ensemble. Supprime l'ensemble si le dernier membre a été retiré.

En savoir plus

- `SSCAN`

  Effectue une itération sur les membres d'un ensemble.

  En savoir plus

- `SUNION`

  Renvoie l'union de plusieurs ensembles.

  En savoir plus

- `SUNIONSTORE`

  Stocke dans une clé l'union de plusieurs ensembles.

  En savoir plus

Commandes d'ensemble trié

- `BZMPOP`

  Retire et renvoie un membre en fonction de son score dans un ou plusieurs ensembles triés. Bloque jusqu'à ce qu'un membre soit disponible autrement. Supprime l'ensemble trié si le dernier élément a été extrait.

  En savoir plus

- `BZPOPMAX`

  Retire et renvoie le membre ayant le score le plus élevé dans ou plusieurs ensembles triés. Bloque jusqu'à ce qu'un membre soit disponible autrement. Supprime l'ensemble trié si le dernier élément a été extrait.

  En savoir plus

- `BZPOPMIN`

Retire et renvoie le membre ayant le score le plus faible dans ou plusieurs ensembles triés. Bloque jusqu'à ce qu'un membre soit disponible autrement. Supprime l'ensemble trié si le dernier élément a été extrait.

En savoir plus

- `ZADD`

  Ajoute un ou plusieurs membres à un ensemble trié ou met à jour leurs scores. Crée la clé si elle n'existe pas.

  En savoir plus

- `ZCARD`

  Renvoie le nombre de membres d'un ensemble trié.

  En savoir plus

- `ZCOUNT`

  Renvoie le nombre de membres d'un ensemble trié dont les scores sont compris dans une plage.

  En savoir plus

- `ZDIFF`

  Renvoie la différence entre plusieurs ensembles triés.

  En savoir plus

- `ZDIFFSTORE`

  Stocke dans une clé la différence entre plusieurs ensembles triés.

  En savoir plus

- `ZINCRBY`

  Incrémente le score d'un membre dans un ensemble trié.

  En savoir plus

- `ZINTER`

  Renvoie l'intersection de plusieurs ensembles triés.

En savoir plus

- `ZINTERCARD`

  Renvoie le nombre de membres de l'intersection de plusieurs ensembles triés.

  En savoir plus

- `ZINTERSTORE`

  Stocke dans une clé l'intersection de plusieurs ensembles triés.

  En savoir plus

- `ZLEXCOUNT`

  Renvoie le nombre de membres d'un ensemble trié compris dans une plage lexicographique.

  En savoir plus

- `ZMPOP`

  Renvoie les membres ayant le score le plus élevé ou le plus faible d'un ou de plusieurs ensembles triés après les avoir retirés. Supprime l'ensemble trié si le dernier membre a été extrait.

  En savoir plus

- `ZMSCORE`

  Renvoie le score d'un ou de plusieurs membres d'un ensemble trié.

  En savoir plus

- `ZPOPMAX`

  Renvoie les membres ayant le score le plus élevé d'un ensemble trié après les avoir retirés. Supprime l'ensemble trié si le dernier membre a été extrait.

  En savoir plus

- `ZPOPMIN`

  Renvoie les membres ayant le score le plus faible d'un ensemble trié après les avoir retirés. Supprime l'ensemble trié si le dernier membre a été extrait.

  En savoir plus

- `ZRANDMEMBER`

  Renvoie un ou plusieurs membres aléatoires d'un ensemble trié.

  En savoir plus
- `ZRANGE`

  Renvoie les membres d'un ensemble trié compris dans une plage d'index.

  En savoir plus
- `ZRANGEBYLEX`

  Renvoie les membres d'un ensemble trié compris dans une plage lexicographique.

  En savoir plus
- `ZRANGEBYSCORE`

  Renvoie les membres d'un ensemble trié compris dans une plage de scores.

  En savoir plus
- `ZRANGESTORE`

  Stocke dans une clé une plage de membres d'un ensemble trié.

  En savoir plus
- `ZRANK`

  Renvoie l'index d'un membre d'un ensemble trié, classé par scores croissants.

  En savoir plus
- `ZREM`

  Retire un ou plusieurs membres d'un ensemble trié. Supprime l'ensemble trié si tous les membres ont été retirés.

  En savoir plus
- `ZREMRANGEBYLEX`

  Retire les membres d'un ensemble trié compris dans une plage lexicographique. Supprime l'ensemble trié si tous les membres ont été retirés.

En savoir plus

- `ZREMRANGEBYRANK`

  Retire les membres d'un ensemble trié compris dans une plage d'index. Supprime l'ensemble trié si tous les membres ont été retirés.

  En savoir plus

- `ZREMRANGEBYSCORE`

  Retire les membres d'un ensemble trié compris dans une plage de scores. Supprime l'ensemble trié si tous les membres ont été retirés.

  En savoir plus

- `ZREVRANGE`

  Renvoie les membres d'un ensemble trié compris dans une plage d'index, dans l'ordre inverse.

  En savoir plus

- `ZREVRANGEBYLEX`

  Renvoie les membres d'un ensemble trié compris dans une plage lexicographique, dans l'ordre inverse.

  En savoir plus

- `ZREVRANGEBYSCORE`

  Renvoie les membres d'un ensemble trié compris dans une plage de scores, dans l'ordre inverse.

  En savoir plus

- `ZREVRANK`

  Renvoie l'index d'un membre d'un ensemble trié, classé par scores décroissants.

  En savoir plus

- `ZSCAN`

  Effectue une itération sur les membres et les scores d'un ensemble trié.

  En savoir plus

- `ZSCORE`

  Renvoie le score d'un membre dans un ensemble trié.

  En savoir plus

- `ZUNION`

  Renvoie l'union de plusieurs ensembles triés.

  En savoir plus

- `ZUNIONSTORE`

  Stocke dans une clé l'union de plusieurs ensembles triés.

  En savoir plus

Commandes de flux

- `XACK`

  Renvoie le nombre de messages qui ont été confirmés avec succès par le membre du groupe de consommateurs d'un flux.

  En savoir plus

- `XADD`

  Ajoute un nouveau message à un flux. Crée la clé si elle n'existe pas.

  En savoir plus

- `XAUTOCLAIM`

  Change ou acquiert la propriété des messages d'un groupe de consommateurs, comme s'ils étaient remis en tant que membre du groupe de consommateurs.

  En savoir plus

- `XCLAIM`

  Change ou acquiert la propriété d'un message d'un groupe de consommateurs, comme s'il était remis en tant que membre du groupe de consommateurs.

En savoir plus

- `XDEL`

  Renvoie le nombre de messages après les avoir retirés d'un flux.

  En savoir plus

- `XGROUP CREATE`

  Crée un groupe de consommateurs.

  En savoir plus

- `XGROUP CREATECONSUMER`

  Crée un consommateur dans un groupe de consommateurs.

  En savoir plus

- `XGROUP DELCONSUMER`

  Supprime un consommateur d'un groupe de consommateurs.

  En savoir plus

- `XGROUP DESTROY`

  Détruit un groupe de consommateurs.

  En savoir plus

- `XGROUP SETID`

  Définit le dernier ID remis d'un groupe de consommateurs.

  En savoir plus

- `XINFO CONSUMERS`

  Renvoie la liste des consommateurs d'un groupe de consommateurs.

  En savoir plus

- `XINFO GROUPS`

  Renvoie la liste des groupes de consommateurs d'un flux.

En savoir plus

- `XINFO STREAM`

  Renvoie des informations sur un flux.

  En savoir plus

- `XLEN`

  Renvoie le nombre de messages dans un flux.

  En savoir plus

- `XPENDING`

  Renvoie les informations et les entrées de la liste des entrées en attente d'un groupe de consommateurs de flux.

  En savoir plus

- `XRANGE`

  Renvoie les messages d'un flux compris dans une plage de IDs.

  En savoir plus

- `XREAD`

  Renvoie des messages provenant de plusieurs flux dont le contenu est IDs supérieur à celui demandé. Bloque jusqu'à ce qu'un message soit disponible autrement.

  En savoir plus

- `XREADGROUP`

  Renvoie les messages nouveaux ou historiques d'un flux pour un consommateur d'un groupe. Bloque jusqu'à ce qu'un message soit disponible autrement.

  En savoir plus

- `XREVRANGE`

  Renvoie les messages d'un flux compris dans un intervalle de valeurs IDs dans l'ordre inverse.

  En savoir plus

- `XTRIM`

  Supprime les messages depuis le début d'un flux.

  En savoir plus

Commandes de chaîne

- `APPEND`

  Ajoute une chaîne à la valeur d'une clé. Crée la clé si elle n'existe pas.

  En savoir plus
- `DECR`

  Décrémente de un la valeur entière d'une clé. Utilise 0 comme valeur initiale si la clé n'existe pas.

  En savoir plus
- `DECRBY`

  Décrémente d'un nombre la valeur entière d'une clé. Utilise 0 comme valeur initiale si la clé n'existe pas.

  En savoir plus
- `GET`

  Renvoie la valeur de chaîne d'une clé.

  En savoir plus
- `GETDEL`

  Renvoie la valeur de chaîne d'une clé après avoir supprimé la clé.

  En savoir plus
- `GETEX`

  Renvoie la valeur de chaîne d'une clé après avoir défini son délai d'expiration.

  En savoir plus
- `GETRANGE`

Renvoie une sous-chaîne de la chaîne stockée dans une clé.

En savoir plus

- `GETSET`

  Renvoie la valeur de chaîne précédente d'une clé après l'avoir définie sur une nouvelle valeur.

  En savoir plus

- `INCR`

  Incrémente de un la valeur entière d'une clé. Utilise 0 comme valeur initiale si la clé n'existe pas.

  En savoir plus

- `INCRBY`

  Incrémente d'un nombre la valeur entière d'une clé. Utilise 0 comme valeur initiale si la clé n'existe pas.

  En savoir plus

- `INCRBYFLOAT`

  Incrémente d'un nombre la valeur à virgule flottante d'une clé. Utilise 0 comme valeur initiale si la clé n'existe pas.

  En savoir plus

- `LCS`

  Recherche la sous-chaîne commune la plus longue.

  En savoir plus

- `MGET`

  Renvoie de manière atomique les valeurs de chaîne d'une ou de plusieurs clés.

  En savoir plus

- `MSET`

  Crée ou modifie de manière atomique les valeurs de chaîne d'une ou de plusieurs clés.

En savoir plus

- `MSETNX`

  Modifie de manière atomique les valeurs de chaîne d'une ou de plusieurs clés uniquement lorsque toutes les clés n'existent pas.

  En savoir plus

- `PSETEX`

  Définit la valeur de chaîne et le délai d'expiration d'une clé en millisecondes. La clé est créée si elle n'existe pas.

  En savoir plus

- `SET`

  Définit la valeur de chaîne d'une clé, en ignorant son type. La clé est créée si elle n'existe pas.

  En savoir plus

- `SETEX`

  Définit la valeur de chaîne et le délai d'expiration d'une clé. Crée la clé si elle n'existe pas.

  En savoir plus

- `SETNX`

  Définit la valeur de chaîne d'une clé uniquement lorsque la clé n'existe pas.

  En savoir plus

- `SETRANGE`

  Remplace une partie d'une valeur de chaîne par une autre selon un décalage. Crée la clé si elle n'existe pas.

  En savoir plus

- `STRLEN`

  Renvoie la longueur d'une valeur de chaîne.

  En savoir plus

- `SUBSTR`

  Renvoie une sous-chaîne d'une valeur de chaîne.

  En savoir plus

Commandes de transaction

- `DISCARD`

  Supprime une transaction.

  En savoir plus
- `EXEC`

  Exécute toutes les commandes d'une transaction.

  En savoir plus
- `MULTI`

  Démarre une transaction.

  En savoir plus

Commandes Valkey et Redis OSS restreintes

Pour offrir une expérience de service géré, ElastiCache restreint l'accès à certaines commandes spécifiques au moteur de cache qui nécessitent des privilèges avancés. Pour les caches exécutant Redis OSS, les commandes suivantes ne sont pas disponibles :

- `acl setuser`
- `acl load`
- `acl save`
- `acl deluser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster addslotsrange`

- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

De plus, les commandes suivantes ne sont pas disponibles pour les caches sans serveur :

- `acl log`
- `client caching`
- `client getredir`
- `client id`
- `client info`
- `client kill`
- `client list`
- `client no-evict`
- `client pause`

- client tracking
- client trackinginfo
- client unblock
- client unpause
- cluster count-failure-reports
- commandlog
- commandlog get
- commandlog help
- commandlog len
- commandlog reset
- fcall
- fcall_ro
- function
- function delete
- function dump
- function flush
- function help
- function kill
- function list
- function load
- function restore
- function stats
- keys
- lastsave
- latency
- latency doctor
- latency graph
- latency help
- latency histogram
- latency history

- `latency latest`
- `latency reset`
- `memory`
- `memory doctor`
- `memory help`
- `memory malloc-stats`
- `memory purge`
- `memory stats`
- `memory usage`
- `monitor`
- `move`
- `object`
- `object encoding`
- `object freq`
- `object help`
- `object idletime`
- `object refcount`
- `pfdebug`
- `pfselftest`
- `psubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`

- `wait`
- `watch`

Commandes Memcached prises en charge

ElastiCache Serverless pour Memcached prend en charge toutes les commandes memcached de la version open source memcached 1.6, à l'exception des suivantes :

- Les connexions client nécessitent le protocole TLS ; par conséquent, le protocole UDP n'est pas pris en charge.
- Le protocole binaire n'est pas pris en charge, car il est officiellement obsolète dans Memcached 1.6.
- Les commandes `GET/GETS` sont limitées à 16 Ko afin d'éviter toute attaque DoS potentielle sur le serveur lors de l'extraction d'un grand nombre de clés.
- La commande `flush_all` différée sera rejetée avec `CLIENT_ERROR`.
- Les commandes qui configurent le moteur ou qui révèlent des informations internes sur l'état du moteur ou les journaux ne sont pas prises en charge. Par exemple :
  - Pour la commande `STATS`, seules `stats` et `stats reset` sont prises en charge. Les autres variantes renverront `ERROR`.
  - `lru / lru_crawler` : modification des paramètres LRU et Crawler LRU
  - `watch` : surveille les journaux du serveur Memcached
  - `verbosity` : configure le niveau de journalisation du serveur
  - `me-` la commande meta debug (me) n'est pas prise en charge

Configuration et limites de Valkey et Redis OSS

Les moteurs Valkey et Redis OSS fournissent chacun un certain nombre de paramètres de configuration, dont certains sont modifiables dans Redis OSS et d'autres non modifiables ElastiCache pour garantir des performances et une fiabilité stables.

Caches sans serveur

Pour les caches sans serveur, les groupes de paramètres ne sont pas utilisés et toutes les configurations de Valkey ou Redis OSS ne sont pas modifiables. Les paramètres Valkey ou Redis OSS suivants sont en place :

Nom	Détails	Description
acl-pubsub-default	allchannels	Autorisations de canal pubsub par défaut pour les utilisateurs ACL sur le cache.
client-output-buffer-limit	normal 0 0 0 pubsub 32mb 8mb 60	Les clients normaux n'ont aucune limite de mémoire tampon. PUB/SUB les clients seront déconnectés s'ils franchissent un backlog de 32 Mo ou s'ils franchissent un backlog de 8 Mo pendant 60 minutes.
client-query-buffer-limit	1 Gio	Taille maximale d'un seul tampon de requête client. En outre, les clients ne peuvent pas émettre de demande comportant plus de 3 999 arguments.
cluster-allow-pubsubshard-when-down	yes	Cela permet au cache d'assurer le trafic pubsub lorsqu'il est partiellement indisponible.
cluster-allow-reads-when-down	yes	Cela permet au cache d'assurer le trafic de lecture lorsqu'il est partiellement indisponible.
cluster-enabled	yes	Tous les caches sans serveur sont en mode cluster activé, ce qui leur permet de partitionner leurs données de manière transparente sur plusieurs partitions de backend. Tous les emplacements sont présentés aux clients comme appartenant à un seul nœud virtuel.
cluster-require-full-coverage	no	Lorsque l'espace de clés est partiellement indisponible (c'est-à-dire qu'au moins un emplacement de hachage est inaccessible), le cache continue à accepter les requêtes pour

Nom	Détails	Description
		la partie de l'espace de clés qui est encore couverte. L'ensemble de l'espace de clés sera toujours « couvert » par un seul nœud virtuel dans `cluster slots`.
`lua-time-limit`	5000	Durée maximale d'exécution d'un script Lua, en millisecondes, avant qu'une action ne soit entreprise pour ElastiCache arrêter le script. En cas *`lua-time-limit`* de dépassement, toutes les commandes Valkey ou Redis OSS peuvent renvoyer une erreur de la forme ____-BUSY. Étant donné que cet état peut interférer avec de nombreuses opérations essentielles de Valkey ou Redis OSS, ElastiCache il émettra d'abord une commande SCRIPT KILL. En cas d'échec, ElastiCache il redémarrera de force Valkey ou Redis OSS.
`maxclients`	65000	Nombre maximal de clients pouvant être connectés au cache simultanément. Les connexions établies par la suite peuvent réussir ou non.
`maxmemory-policy`	`volatile-lru`	Les éléments dont le TTL est défini sont expulsés après une estimation least-recently-used (LRU) lorsque la limite de mémoire d'un cache est atteinte.
`notify-keyspace-events`	(chaîne vide)	Les événements de l'espace de clés ne sont actuellement pas pris en charge sur les caches sans serveur.

Nom	Détails	Description
`port`	Port principal : 6379 Port de lecture : 6380	Les caches sans serveur publient deux ports portant le même nom d'hôte. Le port principal autorise les écritures et les lectures, tandis que le port de lecture autorise les lectures éventuell ement cohérentes à faible latence à l'aide de la commande `READONLY`.
`proto-max -bulk-len`	512 Mio	Taille maximale d'une seule demande d'élément.
`timeout`	0	Les clients ne sont pas déconnectés de force après une période d'inactivité spécifique, mais ils peuvent être déconnectés en état stable à des fins d'équilibrage de charge.

Les limites suivantes s'appliquent également :

Nom	Détails	Description
Taille par cache	5 000 GiB	Quantité maximale de données pouvant être stockées par cache sans serveur.
Taille par emplacement	32 GiO	Taille maximale d'un seul emplacement de hachage Valkey ou Redis OSS. Les clients qui essaient de définir plus de données que cela sur un seul emplacement Valkey ou Redis OSS déclencheront la politique d'expulsion sur cet emplacement et, si aucune clé ne peut être expulsée, recevront une erreur de mémoire insuffisante (). `OOM`
ECPU par cache	15 000 000 ECPU/seconde	ElastiCache Unités de traitement (ECPU) métriques. Le nombre de données ECPUs consommées par vos requêtes dépend du

Nom	Détails	Description
		temps nécessaire au vCPU et de la quantité de données transférées.
ECPU par slot	30 000 à 90 000 ECPU/seconde	Maximum de 30 000 ECPUs/second par emplacement ou de 90 000 ECPUs/second si vous utilisez Read from Replica à l'aide de connexions READONLY.
Arguments par demande	3 999	Nombre maximum d'arguments par demande. Les clients qui envoient un plus grand nombre d'arguments par demande recevront un message d'erreur.
Longueur du nom de clé	4 Kio	La taille maximale d'une seule clé Valkey ou Redis OSS ou d'un seul nom de canal. Les clients faisant référence à des clés dont la taille est supérieure à cette valeur recevront une erreur.
Taille de script Lua	4 Mio	Taille maximale d'un seul script Lua Valkey ou Redis OSS. Les tentatives de chargement d'un script Lua dont la taille est supérieure à cette valeur provoqueront un message d'erreur.

Clusters auto-conçus

Pour les clusters auto-conçus, consultez Paramètres OSS Valkey et Redis pour obtenir les valeurs par défaut des paramètres de configuration et connaître celles qui sont configurables. Les valeurs par défaut sont généralement recommandées, sauf si vous devez les remplacer dans un cas d'utilisation spécifique.

IPv6 exemples de clients pour Valkey, Memcached et Redis OSS

ElastiCache est compatible avec Valkey, Memcached et Redis OSS. Cela signifie que les clients qui prennent en charge IPv6 les connexions devraient pouvoir se connecter à des clusters IPv6 activés

ElastiCache pour Memcached. Certaines mises en garde méritent d'être prises en compte lors de l'interaction avec des ressources IPv6 activées.

Vous pouvez consulter le billet de blog consacré aux meilleures pratiques pour les clients Valkey et Redis sur le blog de AWS base de données pour obtenir des recommandations sur la configuration des clients Valkey et Redis OSS pour les ressources. ElastiCache

Vous trouverez ci-dessous les meilleures pratiques pour interagir avec les ElastiCache ressources IPv6 activées dans les bibliothèques clientes open source couramment utilisées.

Clients validés avec Valkey et Redis OSS

ElastiCache est compatible avec Valkey et Redis OSS open source. Cela signifie que les clients Valkey et Redis OSS open source qui prennent en charge IPv6 les connexions devraient pouvoir se connecter aux clusters Redis OSS IPv6 activés ElastiCache . En outre, plusieurs des clients Python et Java les plus populaires ont été spécifiquement testés et validés pour fonctionner avec toutes les configurations de type réseau prises en charge (IPv4 uniquement, IPv6 uniquement et Dual Stack)

Les clients suivants ont été spécifiquement validés pour fonctionner avec toutes les configurations de type de réseau prises en charge pour Valkey et Redis OSS.

Clients validés :

- Redis Py () : 4,1.2
- Lettuce : version : 6.1.6.RELEASE
- Jedis : version : 3.6.0

Bonnes pratiques pour les clients (Valkey et Redis OSS)

Découvrez les meilleures pratiques pour les scénarios courants et suivez les exemples de code de certaines des bibliothèques clientes open source Valkey et Redis OSS les plus populaires (redis-py et Lettuce) PHPRedis, ainsi que les meilleures pratiques pour interagir avec les ElastiCache ressources des bibliothèques clientes Memcached open source couramment utilisées.

Rubriques

- Grand nombre de connexions (Valkey et Redis OSS)
- Découverte des clients du cluster et ralentissement exponentiel (Valkey et Redis OSS)
- Configurer un délai d'expiration côté client (Valkey et Redis OSS)

- Configurer un délai d'inactivité côté serveur (Valkey et Redis OSS)
- Scripts Lua
- Stockage d'objets composites de grande taille (Valkey et Redis OSS)
- Configuration du client Lettuce (Valkey et Redis OSS)
- Configuration d'un protocole préféré pour les clusters à double pile (Valkey et Redis OSS)

Grand nombre de connexions (Valkey et Redis OSS)

Les caches sans serveur et individuels ElastiCache pour les nœuds Redis OSS prennent en charge jusqu'à 65 000 connexions client simultanées. Toutefois, pour optimiser les performances, nous recommandons que les applications client ne fonctionnent pas constamment à ce niveau de connexions. Valkey et Redis OSS ont chacun un processus à thread unique basé sur une boucle d'événements dans laquelle les demandes des clients entrantes sont traitées de manière séquentielle. Cela signifie que le temps de réponse d'un client donné s'allonge à mesure que le nombre de clients connectés augmente.

Vous pouvez prendre les mesures suivantes pour éviter de rencontrer un goulot d'étranglement de connexion sur un serveur Valkey ou Redis OSS :

- Effectuez des opérations de lecture à partir de réplicas en lecture. Cela peut être fait en désactivant les points de terminaison du ElastiCache lecteur en mode cluster ou en utilisant des répliques pour les lectures en mode cluster activé, y compris un cache sans serveur.
- Répartissez le trafic d'écriture sur plusieurs nœuds primaires. Vous pouvez effectuer cette opération de deux façons. Vous pouvez utiliser un cluster Valkey ou Redis OSS multipartitionné avec un client compatible avec le mode cluster. Vous pouvez également écrire sur plusieurs nœuds primaires en mode cluster désactivé avec le partitionnement côté client. Ce processus est automatique dans un cache sans serveur.
- Utilisez un groupe de connexions lorsqu'il est disponible dans votre bibliothèque client.

En général, la création d'une connexion TCP est une opération coûteuse en termes de calcul par rapport aux commandes classiques de Valkey ou Redis OSS. Par exemple, le traitement d'une SET/GET demande est d'un ordre de grandeur plus rapide lors de la réutilisation d'une connexion existante. L'utilisation d'un groupe de connexions client de taille limitée réduit les frais de gestion des connexions. Elle limite également le nombre de connexions entrantes simultanées provenant de l'application client.

L'exemple de code suivant PHPRedis montre qu'une nouvelle connexion est créée pour chaque nouvelle demande utilisateur :

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
 //ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Nous avons comparé ce code en boucle sur une instance Amazon Elastic Compute Cloud (Amazon EC2) connectée à un nœud Graviton2 (m6g.2xlarge) pour Redis OSS. ElastiCache Nous avons placé le client et le serveur dans la même zone de disponibilité. La latence moyenne de l'ensemble de l'opération était de 2,82 millisecondes.

Lorsque nous avons mis à jour le code et utilisé des connexions persistantes et un groupe de connexions, la latence moyenne de l'ensemble de l'opération était de 0,21 milliseconde :

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
 // ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Configurations redis.ini requises :

- `redis.pconnect.pooling_enabled=1`
- `redis.pconnect.connection_limit=10`

Le code suivant est un exemple de groupe de connexions Redis-py :

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
conn.set(key, value)
```

Le code suivant est un exemple de groupe de connexions Lettuce :

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
 ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
 GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
 RedisCommands syncCommands = connection.sync();
 syncCommands.set(key, value);
}
```

Découverte des clients du cluster et ralentissement exponentiel (Valkey et Redis OSS)

Lorsque vous vous connectez à un cluster ElastiCache Valkey ou Redis OSS en mode cluster activé, la bibliothèque cliente correspondante doit être consciente du cluster. Les clients doivent obtenir une carte des emplacements de hachage sur les nœuds correspondants du cluster afin d'envoyer des demandes aux nœuds appropriés et d'éviter la surcharge de performances liée à la gestion des redirections du cluster. Par conséquent, le client doit découvrir la liste complète des emplacements et des nœuds mappés dans deux cas différents :

- Le client est initialisé et doit renseigner la configuration initiale des emplacements.
- Une redirection MOVED est reçue du serveur, par exemple en cas de basculement lorsque tous les emplacements desservis par l'ancien nœud primaire sont contrôlés par le réplica, ou en cas de repartitionnement lorsque les emplacements sont déplacés du nœud primaire source vers le nœud primaire cible.

La découverte du client se fait généralement en émettant une commande CLUSTER SLOT ou CLUSTER NODE au serveur Valkey ou Redis OSS. Nous recommandons la méthode CLUSTER SLOT, car elle renvoie au client l'ensemble des plages d'emplacements ainsi que les nœuds primaires et de réplica associés. Cette méthode ne nécessite pas d'analyse supplémentaire de la part du client. Elle est également plus efficace.

En fonction de la topologie du cluster, la taille de la réponse à la commande CLUSTER SLOT peut varier en fonction de la taille du cluster. Les clusters de plus grande taille dotés d'un plus grand nombre de nœuds produisent une réponse plus longue. Par conséquent, il est important de veiller à ce que le nombre de clients qui effectuent la découverte de la topologie du cluster n'augmente pas de manière illimitée. Par exemple, lorsque l'application client démarre ou perd la connexion avec le serveur et doit effectuer une découverte de clusters, une erreur courante est qu'elle lance plusieurs

demandes de reconnexion et de découverte sans ajouter de backoff exponentiel lors d'une nouvelle tentative. Cela peut empêcher le serveur Valkey ou Redis OSS de répondre pendant une période prolongée, avec une utilisation du processeur à 100 %. La panne est prolongée si chaque commande CLUSTER SLOT doit traiter un grand nombre de nœuds dans le bus du cluster. Nous avons observé plusieurs pannes de clients dans le passé en raison de ce comportement dans différents langages, notamment Python (redis-py-cluster) et Java (Lettuce et Redisson).

Dans un cache sans serveur, de nombreux problèmes sont automatiquement atténués, car la topologie du cluster annoncée est statique et comprend deux entrées : un point de terminaison d'écriture et un point de terminaison de lecture. De plus, la découverte de clusters est automatiquement répartie sur plusieurs nœuds lors de l'utilisation du point de terminaison du cache. Les recommandations suivantes restent toutefois utiles.

Pour atténuer l'impact provoqué par un afflux soudain de demandes de connexion et de découverte, nous recommandons ce qui suit :

- Implémentez un groupe de connexions client de taille limitée pour limiter le nombre de connexions entrantes simultanées en provenance de l'application client.
- Lorsque le client se déconnecte du serveur en raison d'un délai d'expiration, réessayez en ajoutant un backoff exponentiel avec instabilité. Cela permet d'éviter que plusieurs clients ne surchargent le serveur en même temps.
- Consultez Recherche de points de terminaison de connexion dans ElastiCache pour trouver le point de terminaison de cluster afin d'effectuer la découverte de clusters. En agissant ainsi, vous répartissez la charge de découverte sur tous les nœuds du cluster (jusqu'à 90) au lieu de toucher quelques nœuds de départ codés en dur du cluster.

Voici quelques exemples de code pour la logique de réessai exponentielle dans redis-py,, et Lettuce. PHPRedis

Exemple de logique avec backoff 1 : redis-py

redis-py possède un mécanisme de nouvelle tentative intégré qui permet d'effectuer une nouvelle tentative immédiatement après un échec. Ce mécanisme peut être activé via l'`retry_on_timeout`argument fourni lors de la création d'un objet Redis OSS. L'exemple ci-dessous illustre un mécanisme de nouvelle tentative personnalisé avec backoff exponentiel et instabilité. Nous avons soumis une demande d'extraction pour implémenter de manière native le backoff exponentiel dans redis-py (#1494). À l'avenir, il ne sera peut-être plus nécessaire de procéder à une implémentation manuelle.

```
def run_with_backoff(function, retries=5):
base_backoff = 0.1 # base 100ms backoff
max_backoff = 10 # sleep for maximum 10 seconds
tries = 0
while True:
try:
  return function()
except (ConnectionError, TimeoutError):
  if tries >= retries:
 raise
  backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
  print(f"sleeping for {backoff:.2f}s")
  sleep(backoff)
  tries += 1
```

Vous pouvez ensuite utiliser le code suivant pour définir une valeur :

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

En fonction de votre charge de travail, vous souhaiterez peut-être modifier la valeur de backoff de base de 1 seconde à quelques dizaines ou centaines de millisecondes pour les charges de travail sensibles à la latence.

Exemple de logique de rétrogradation 2 : PHPRedis

PHPRedis possède un mécanisme intégré qui permet de réessayer 10 fois (non configurable) au maximum. Il est possible de configurer un délai entre les tentatives (avec une instabilité à partir de la deuxième tentative). Pour plus d'informations, consultez l'exemple de code ci-après. Nous avons soumis une pull request pour implémenter nativement le backoff exponentiel dans PHPredis (#1986) qui a depuis été fusionnée et documentée. Pour ceux qui utilisent la dernière version de PHPRedis, il ne sera pas nécessaire de l'implémenter manuellement, mais nous avons inclus ici la référence pour ceux des versions précédentes. L'exemple de code suivant permet de configurer le délai du mécanisme de nouvelle tentative :

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
```

```
 return; // ERROR: connection failed
}
$client->set($key, $value);
```

Exemple de logique avec backoff 3 : Lettuce

Lettuce possède des mécanismes de nouvelle tentative intégrés basés sur les stratégies de backoff exponentiel décrites dans l'article Backoff exponentiel et instabilité (langue française non garantie). Voici un extrait de code qui illustre l'approche avec instabilité complète :

```
public static void main(String[] args)
{
 ClientResources resources = null;
 RedisClient client = null;

 try {
  resources = DefaultClientResources.builder()
    .reconnectDelay(Delay.fullJitter(
   Duration.ofMillis(100),     // minimum 100 millisecond delay
   Duration.ofSeconds(5),      // maximum 5 second delay
   100, TimeUnit.MILLISECONDS) // 100 millisecond base
  ).build();

  client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
  client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
 5 second command timeout
 .build());

     // use the connection pool from above example
 } finally {
  if (connection != null) {
   connection.close();
  }

  if (client != null){
   client.shutdown();
  }

  if (resources != null){
   resources.shutdown();
```

```
  }

 }
}
```

Configurer un délai d'expiration côté client (Valkey et Redis OSS)

Configuration du délai d'expiration côté client

Configurez le délai d'expiration côté client de manière appropriée pour laisser au serveur suffisamment de temps pour traiter la demande et générer la réponse. Il pourra également procéder à une interruption immédiate si la connexion au serveur ne peut pas être établie. Certaines commandes Valkey ou Redis OSS peuvent être plus coûteuses en termes de calcul que d'autres. Par exemple, les scripts Lua ou MULTI/EXEC les transactions contenant plusieurs commandes qui doivent être exécutées de manière atomique. En général, il est recommandé d'augmenter le délai d'expiration côté client pour éviter que le client n'expire avant de recevoir la réponse du serveur, notamment dans les cas suivants :

- Exécution de commandes sur plusieurs clés
- Exécution de MULTI/EXEC transactions ou de scripts Lua composés de plusieurs commandes Valkey ou Redis OSS individuelles
- Lecture de valeurs élevées
- Exécution d'opérations de blocage telles que BLPOP

Dans le cas d'une opération de blocage telle que BLPOP, la bonne pratique consiste à régler le délai d'expiration de la commande sur un nombre inférieur au délai d'expiration du socket.

Voici des exemples de code pour implémenter un délai d'attente côté client dans redis-py et Lettuce. PHPRedis

Exemple de configuration du délai d'expiration 1 : redis-py

Voici un exemple de code avec redis-py :

```
# connect to Redis server with a 100 millisecond timeout
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
```

```
print(res)                      # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                      # less than the 2 second socket timeout
print(res)
```

Exemple de configuration de temporisation 2 : PHPRedis

Voici un exemple de code avec PHPRedis :

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
 return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                      // if there is a connection error

$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                   // less than the 2 second socket timeout
```

Exemple de configuration du délai d'expiration 3 : Lettuce

Voici un exemple de code avec Lettuce :

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
 RedisClient client = null;
 StatefulRedisConnection<String, String> connection = null;
 try {
  client = RedisClient.create(RedisURI.create(HOST, PORT));
  client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
 2 second command timeout
```

```
  .build());

   // use the connection pool from above example

   commands.set("key", "value"); // will timeout after 2 seconds
   commands.blpop(1, "list"); // BLPOP with 1 second timeout
  } finally {
   if (connection != null) {
    connection.close();
   }

   if (client != null){
    client.shutdown();
   }
  }
 }
```

Configurer un délai d'inactivité côté serveur (Valkey et Redis OSS)

Nous avons observé des cas où l'application d'un client compte un grand nombre de clients inactifs connectés, mais n'envoie pas activement de commandes. Dans de tels scénarios, vous pouvez épuiser les 65 000 connexions avec un nombre élevé de clients inactifs. Pour éviter de tels scénarios, configurez le paramètre de délai d'expiration de manière appropriée sur le serveur via Paramètres OSS Valkey et Redis. Cela garantit que le serveur déconnecte activement les clients inactifs afin d'éviter une augmentation du nombre de connexions. Ce paramètre n'est pas disponible sur les caches sans serveur.

Scripts Lua

Valkey et Redis OSS prennent en charge plus de 200 commandes, y compris celles permettant d'exécuter des scripts Lua. Cependant, en ce qui concerne les scripts Lua, plusieurs pièges peuvent affecter la mémoire et la disponibilité de Valkey ou Redis OSS.

Scripts Lua non paramétrés

Chaque script Lua est mis en cache sur le serveur Valkey ou Redis OSS avant son exécution. Les scripts Lua non paramétrés sont uniques, ce qui peut amener le serveur Valkey ou Redis OSS à stocker un grand nombre de scripts Lua et à consommer davantage de mémoire. Pour atténuer ce problème, assurez-vous que tous les scripts Lua sont paramétrés et exécutez régulièrement la commande SCRIPT FLUSH pour nettoyer les scripts Lua mis en cache, si nécessaire.

Sachez également que les clés doivent être fournies. Si aucune valeur n'est fournie pour le paramètre KEY, le script échouera. Par exemple, cela ne fonctionnera pas :

```
serverless-test-lst4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return "Hello
 World"' 0
(error) ERR Lua scripts without any input keys are not supported.
```

Cela fonctionnera :

```
serverless-test-lst4hg.serverless.use1.cache.amazonaws.com:6379> eval 'return
 redis.call("get", KEYS[1])' 1 mykey-2
"myvalue-2"
```

L'exemple suivant illustre l'utilisation des scripts paramétrés. Le premier exemple d'approche non paramétrée, qui n'est pas recommandée, donne lieu à trois scripts Lua mis en cache différents :

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Utilisez plutôt le modèle suivant pour créer un script unique capable d'accepter les paramètres transmis :

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Scripts Lua de longue durée

Les scripts Lua peuvent exécuter plusieurs commandes de manière atomique. Leur exécution peut donc prendre plus de temps qu'une commande Valkey ou Redis OSS normale. Si le script Lua exécute uniquement des opérations en lecture seule, vous pouvez l'arrêter en plein milieu. Cependant, dès que le script Lua exécute une opération d'écriture, il devient impossible de l'arrêter et il doit être s'exécuter jusqu'à la fin. Un script Lua de longue durée qui est en train de muter peut empêcher le serveur Valkey ou Redis OSS de répondre pendant une longue période. Pour atténuer ce problème, évitez les scripts Lua de longue durée et testez le script dans un environnement de pré-production.

Script Lua avec écritures en mode furtif

Un script Lua peut continuer à écrire de nouvelles données dans Valkey ou Redis OSS de plusieurs manières, même lorsque Valkey ou Redis OSS est terminé : `maxmemory`

- Le script démarre lorsque le serveur Valkey ou Redis OSS se trouve en dessous et contient plusieurs `maxmemory` opérations d'écriture à l'intérieur
- La première commande d'écriture du script ne consomme pas de mémoire (comme DEL), puis les autres opérations d'écriture consomment de la mémoire
- Vous pouvez atténuer ce problème en configurant une politique d'expulsion appropriée dans le serveur Valkey ou Redis OSS autre que. `noeviction` Cela permet à Redis OSS d'expulser des éléments et de libérer de la mémoire entre les scripts Lua.

Stockage d'objets composites de grande taille (Valkey et Redis OSS)

Dans certains scénarios, une application peut stocker de gros éléments composites dans Valkey ou Redis OSS (comme un jeu de données de hachage de plusieurs Go). Cette pratique n'est pas recommandée car elle entraîne souvent des problèmes de performance dans Valkey ou Redis OSS. Par exemple, le client peut exécuter une commande HGETALL pour récupérer l'intégralité de la collection de hachage de plusieurs Go. Cela peut générer une pression de mémoire importante sur le serveur Valkey ou Redis OSS qui met en mémoire tampon le gros élément dans la mémoire tampon de sortie du client. En outre, pour la migration d'emplacements en mode cluster, les emplacements contenant des éléments dont la taille sérialisée est supérieure à 256 Mo ElastiCache ne sont pas migrés.

Pour résoudre les problèmes liés aux éléments de grande taille, suivez les recommandations ci-dessous :

- Divisez l'élément composite de grande taille en plusieurs éléments de plus petite taille. Par exemple, divisez une collection de hachage de grande taille en champs clé-valeur individuels avec un schéma de nom de clé qui reflète la collection de manière appropriée, en utilisant par exemple un préfixe commun dans le nom de clé pour identifier la collection d'éléments. Si vous devez accéder à plusieurs champs de la même collection de manière atomique, vous pouvez utiliser la commande MGET pour récupérer plusieurs valeurs-clés dans la même commande.
- Si vous avez évalué toutes les options et que vous ne parvenez toujours pas à diviser le jeu de données de la collection de grande taille, essayez d'utiliser des commandes qui fonctionnent sur un sous-ensemble des données de la collection plutôt que sur l'ensemble de la collection. Évitez les cas d'utilisation qui vous obligent à récupérer de manière atomique l'intégralité de la collection de

plusieurs Go dans la même commande. L'utilisation de commandes HGET ou HMGET au lieu de HGETALL sur les collections de hachage en est un exemple.

Configuration du client Lettuce (Valkey et Redis OSS)

Cette section décrit les options de configuration recommandées pour Java et Lettuce, ainsi que leur application aux ElastiCache clusters.

Les recommandations de cette section ont été testées avec la version 6.2.2 de Lettuce.

Rubriques

- Exemple : configuration de laitue pour le mode cluster, TLS activé
- Exemple : configuration de laitue pour le mode cluster désactivée, TLS activé

TTL du cache DNS de Java

La machine virtuelle Java (JVM) met en cache les recherches de nom DNS. Lorsque la JVM convertit un nom d'hôte en adresse IP, elle met l'adresse IP en cache pendant une période spécifiée, connue sous le nom de time-to-live(TTL).

Le choix d'une valeur TTL est un compromis entre la latence et la réactivité au changement. En TTLs raccourcissant, les résolveurs DNS détectent plus rapidement les mises à jour du DNS du cluster. Cela peut permettre à votre application de réagir plus rapidement aux remplacements ou à d'autres flux de travail auxquels votre cluster est soumis. Toutefois, si la TTL est trop faible, le volume de requêtes augmente, ce qui peut accentuer la latence de votre application. Bien qu'il n'y ait pas de valeur TTL correcte, lorsque vous la configurez, vous pouvez déterminer le délai d'attente maximal avant qu'une modification prenne effet.

Comme ElastiCache les nœuds utilisent des entrées de nom DNS susceptibles de changer, nous vous recommandons de configurer votre JVM avec un TTL faible de 5 à 10 secondes. Ainsi, lorsque l'adresse IP d'un nœud change, votre application peut recevoir et utiliser la nouvelle adresse IP de la ressource en interrogeant l'entrée DNS.

Dans certaines configurations Java, la durée de vie par défaut de la JVM est définie de façon à ce que la JVM n'actualise jamais les entrées DNS tant qu'elle n'est pas redémarrée.

Pour plus de détails sur la façon de définir la durée de vie de votre JVM, consultez Comment définir la durée de vie de la JVM.

Version de Lettuce

Nous recommandons Lettuce version 6.2.2 ou ultérieure.

Points de terminaison

Lorsque vous utilisez des clusters activés en mode cluster, définissez `redisUri` sur le point de terminaison de configuration du cluster. La recherche DNS pour cet URI renvoie une liste de tous les nœuds disponibles dans le cluster et est résolue de manière aléatoire vers l'un d'entre eux lors de l'initialisation du cluster. Pour plus de détails sur le fonctionnement de l'actualisation topologique, reportez-vous à la section dynamicRefreshResourcessuivante de cette rubrique.

SocketOption

Activez KeepAlive. L'activation de cette option réduit la nécessité de gérer les échecs de connexion lors de l'exécution des commandes.

Assurez-vous de définir le Délai de connexion en fonction des exigences de votre application et de votre charge de travail. Pour plus d'informations, consultez la section Délais plus loin dans cette rubrique.

ClusterClientOption: options client activées en mode cluster

Activez AutoReconnecten cas de perte de connexion.

Configurez CommandTimeout. Pour plus de détails, consultez la section Délais plus loin dans cette rubrique.

Définissez nodeFilter pour exclure les nœuds défaillants de la topologie. Lettuce enregistre tous les nœuds présents dans la sortie « nœuds de cluster » (y compris les nœuds ayant le statut PFAIL/FAIL) dans les « partitions » du client. Au cours du processus de création de la topologie du cluster, celle-ci tente de se connecter à tous les nœuds de partition. Ce comportement de Lettuce qui consiste à ajouter des nœuds défaillants peut provoquer des erreurs de connexion (ou des avertissements) lorsque des nœuds sont remplacés pour une raison quelconque.

Par exemple, une fois qu'un basculement est terminé et que le cluster démarre le processus de restauration, alors que la topologie du cluster est actualisée, la carte des nœuds de bus du cluster indique brièvement que le nœud inférieur est listé en tant que nœud FAIL, avant qu'il ne soit entièrement supprimé de la topologie. Pendant cette période, le client Lettuce considère qu'il s'agit d'un nœud sain et s'y connecte en permanence. Cela provoque un échec une fois que les nouvelles tentatives sont épuisées.

Par exemple :

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
    ... // other options
    .nodeFilter(it ->
        ! (it.is(RedisClusterNode.NodeFlag.FAIL)
        || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
        || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
        || it.is(RedisClusterNode.NodeFlag.NOADDR)))
    .validateClusterNodeMembership(false)
    .build();
redisClusterClient.setOptions(clusterClientOptions);
```

> **Note**
>
> Il est préférable d'utiliser le filtrage des nœuds lorsqu'il est DynamicRefreshSources défini sur true. Sinon, si la vue topologique provient d'un seul nœud source problématique, qui constate la défaillance d'un nœud primaire d'une partition, elle filtrera ce nœud primaire et les emplacements ne seront donc pas couverts. Le fait d'avoir plusieurs nœuds de départ (lorsque DynamicRefreshSources c'est vrai) réduit le risque de ce problème, car au moins certains nœuds de départ devraient avoir une vue topologique mise à jour après un basculement avec le nœud principal récemment promu.

ClusterTopologyRefreshOptions: Options permettant de contrôler l'actualisation de la topologie du cluster du client activé en mode cluster

> **Note**
>
> Les clusters en mode cluster désactivé ne prennent pas en charge les commandes de découverte de clusters et ne sont pas compatibles avec toutes les fonctionnalités de découverte de topologie dynamique de clients.
> Le mode cluster désactivé ElastiCache n'est pas compatible avec Lettuce`MasterSlaveTopologyRefresh`. Au lieu de cela, pour le mode cluster désactivé, vous pouvez configurer un `StaticMasterReplicaTopologyProvider` et fournir les points de terminaison de lecture et d'écriture du cluster.

Pour plus d'informations sur la connexion à des clusters en mode cluster désactivé, veuillez consulter Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console).
Si vous souhaitez utiliser la fonctionnalité de découverte de topologie dynamique de Lettuce, vous pouvez créer un cluster en mode cluster activé avec la même configuration de partitions que votre cluster existant. Toutefois, pour les clusters en mode cluster activé, nous vous recommandons de configurer au moins 3 partitions avec au moins 1 réplica pour prendre en charge un basculement rapide.

Activez enablePeriodicRefresh. Cela active les mises à jour périodiques de la topologie du cluster, afin que le client puisse mettre à jour la topologie du cluster dans les intervalles de refreshPeriod (par défaut : 60 secondes). Lorsqu'il est désactivé, le client met à jour la topologie du cluster uniquement lorsque des erreurs se produisent s'il tente d'exécuter des commandes sur le cluster.

Lorsque cette option est activée, vous pouvez réduire la latence associée à l'actualisation de la topologie du cluster en ajoutant cette tâche à une tâche en arrière-plan. Bien que l'actualisation de la topologie soit effectuée en arrière-plan, elle peut être quelque peu lente pour les clusters comportant de nombreux nœuds. Cela est dû au fait que tous les nœuds sont interrogés afin de connaître leurs vues et d'obtenir la vue de cluster la plus récente. Si vous gérez un cluster de grande taille, vous souhaiterez peut-être augmenter la période.

Activez enableAllAdaptiveRefreshTriggers. Cela active une actualisation adaptative de la topologie qui utilise tous les déclencheurs : MOVED_REDIRECT, ASK_REDIRECT, PERSISTENT_RECONNECTS, UNCOVERED_SLOT, UNKNOWN_NODE. Les déclencheurs d'actualisation adaptatifs initient les mises à jour de la vue topologique en fonction des événements qui se produisent lors des opérations du cluster Valkey ou Redis OSS. L'activation de cette option entraîne une actualisation immédiate de la topologie lorsque l'un des déclencheurs précédents s'active. La fréquence des actualisations déclenchées adaptatives est limitée grâce à un délai d'attente, car les événements peuvent se produire à grande échelle (délai d'attente par défaut entre les mises à jour : 30).

Activez closeStaleConnections. Cela permet de fermer les connexions obsolètes lors de l'actualisation de la topologie du cluster. Il n'entre en vigueur que si ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled () est vrai. Lorsqu'elle est activée, le client peut fermer les connexions obsolètes et en créer d'autres en arrière-plan. L'activation de cette option réduit la nécessité de gérer les échecs de connexion lors de l'exécution des commandes.

Activez dynamicRefreshResources. Nous recommandons de l'activer dynamicRefreshResources pour les petits clusters et de la désactiver pour les grands clusters. dynamicRefreshResourcespermet de découvrir les nœuds de cluster à partir du nœud d'origine fourni (par exemple, point de terminaison de configuration du cluster). Il utilise tous les nœuds détectés en tant que sources pour actualiser la topologie du cluster.

L'actualisation dynamique interroge tous les nœuds détectés pour la topologie du cluster et tente de choisir la vue de cluster la plus précise. S'il est défini sur false, seuls les nœuds de départ initiaux sont utilisés comme sources pour la détection de la topologie, et le nombre de clients est obtenu uniquement pour les nœuds de départ initiaux. Lorsqu'il est désactivé, si le point de terminaison de la configuration du cluster est résolu en un nœud défaillant, la tentative d'actualisation de la vue du cluster échoue et entraîne des exceptions. Ce scénario peut se produire, car il faut un certain temps pour que l'entrée d'un nœud défaillant soit supprimée du point de terminaison de configuration du cluster. Par conséquent, le point de terminaison de la configuration peut toujours être résolu de manière aléatoire et brève en un nœud défaillant.

Cependant, lorsqu'il est activé, nous utilisons tous les nœuds de cluster reçus depuis la vue du cluster pour demander leur vue actuelle. Étant donné que nous éliminons les nœuds défaillants de cette vue, l'actualisation de la topologie sera réussie. Toutefois, lorsque dynamicRefreshSources c'est vrai, Lettuce interroge tous les nœuds pour obtenir la vue du cluster, puis compare les résultats. L'opération peut donc être coûteuse pour les clusters comportant un grand nombre de nœuds. Nous vous suggérons de désactiver cette fonctionnalité pour les clusters comportant de nombreux nœuds.

```
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
    .enableAllAdaptiveRefreshTriggers()
    .enablePeriodicRefresh()
    .dynamicRefreshSources(true)
    .build();
```

ClientResources

Configurez DnsResolver avec DirContextDnsResolver. Le résolveur DNS est basé sur com.sun.jndi.dns de Java. DnsContextFactory.

Configurez reconnectDelay avec un backoff exponentiel et une instabilité complète. Lettuce possède des mécanismes de relance intégrés basés sur des stratégies de backoff exponentiel. Pour plus de détails, consultez Exponential Backoff and Jitter sur le AWS blog d'architecture. Pour plus d'informations sur l'importance d'adopter une nouvelle stratégie d'interruption, consultez les sections

sur la logique d'interruption du billet de blog consacré aux meilleures pratiques sur le AWS blog de base de données.

```
ClientResources clientResources = DefaultClientResources.builder()
   .dnsResolver(new DirContextDnsResolver())
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),     // minimum 100 millisecond delay
            Duration.ofSeconds(10),      // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .build();
```

Délais

Utilisez une valeur de délai de connexion inférieure à celle de votre commande. Lettuce utilise l'établissement d'une connexion différée. Ainsi, si le délai d'expiration de la connexion est supérieur à celui de la commande, vous pouvez connaître des échecs persistants après une actualisation de la topologie, si Lettuce essaie de se connecter à un nœud défectueux et si le délai de la commande est toujours dépassé.

Utilisez un délai de commande dynamique pour différentes commandes. Nous vous recommandons de définir le délai de commande en fonction de la durée attendue de la commande. Par exemple, utilisez un délai plus long pour les commandes qui itèrent sur plusieurs clés, comme les scripts FLUSHDB, FLUSHALL, KEYS, SMEMBERS ou Lua. Utilisez des délais plus courts pour les commandes à clé unique, telles que SET, GET et HSET.

> **Note**
>
> Les délais configurés dans l'exemple suivant concernent les tests qui ont exécuté des commandes SET/GET avec des clés et des valeurs d'une longueur maximale de 20 octets. Le temps de traitement peut être supérieur lorsque les commandes sont complexes ou que les clés et les valeurs sont plus grandes. Vous devez définir les délais en fonction du cas d'utilisation de votre application.

```
private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);
```

```
SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();


class DynamicClusterTimeout extends TimeoutSource {
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.CLUSTER)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
          .build();

    private final Duration defaultCommandTimeout;
    private final Duration metaCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
.timeoutSource(
    new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
.build();
```

Exemple : configuration de laitue pour le mode cluster, TLS activé

> **Note**
>
> Dans l'exemple suivant, les délais d'attente concernent les tests qui ont exécuté des SET/GET commandes avec des clés et des valeurs d'une longueur maximale de 20 octets. Le temps de traitement peut être supérieur lorsque les commandes sont complexes ou que les clés et les valeurs sont plus grandes. Vous devez définir les délais en fonction du cas d'utilisation de votre application.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
 cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),     // minimum 100 millisecond delay
            Duration.ofSeconds(10),      // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
```

```
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.CLUSTER)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
          .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
 META_COMMAND_TIMEOUT))
     .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
    .enableAllAdaptiveRefreshTriggers()
    .enablePeriodicRefresh()
    .dynamicRefreshSources(true)
    .build();
```

```
// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .keepAlive(true)
    .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
    .topologyRefreshOptions(topologyOptions)
    .socketOptions(socketOptions)
    .autoReconnect(true)
    .timeoutOptions(timeoutOptions)
    .nodeFilter(it ->
        ! (it.is(RedisClusterNode.NodeFlag.FAIL)
        || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
        || it.is(RedisClusterNode.NodeFlag.NOADDR)))
    .validateClusterNodeMembership(false)
    .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Exemple : configuration de laitue pour le mode cluster désactivée, TLS activé

> **Note**
>
> Dans l'exemple suivant, les délais d'attente concernent les tests qui ont exécuté des SET/GET commandes avec des clés et des valeurs d'une longueur maximale de 20 octets. Le temps de traitement peut être supérieur lorsque les commandes sont complexes ou que les clés et les valeurs sont plus grandes. Vous devez définir les délais en fonction du cas d'utilisation de votre application.

```
// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

 RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
    DefaultClientResources.builder()
        .dnsResolver(new DirContextDnsResolver())
        .reconnectDelay(
            Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(10), // maximum 10 second delay
                100,
                TimeUnit.MILLISECONDS)) // 100 millisecond base
        .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
     private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
          .add(CommandType.FLUSHDB)
          .add(CommandType.FLUSHALL)
          .add(CommandType.INFO)
          .add(CommandType.KEYS)
          .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
```

```
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
     .build();

final SocketOptions socketOptions =
    SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

 ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);
```

Configuration d'un protocole préféré pour les clusters à double pile (Valkey et Redis OSS)

Pour les clusters Valkey ou Redis OSS activés en mode cluster, vous pouvez contrôler le protocole que les clients utiliseront pour se connecter aux nœuds du cluster avec le paramètre IP Discovery. Le paramètre IP Discovery peut être défini sur IPv4 ou IPv6.

Pour les clusters Valkey ou Redis OSS, le paramètre de découverte IP définit le protocole IP utilisé dans les sorties des slots (), des clusters shards () et des nœuds de cluster (). Ces commandes sont utilisées par les clients pour découvrir la topologie du cluster. Les clients utilisent les commandes IPs in these pour se connecter aux autres nœuds du cluster.

La modification de la découverte d'adresses IP n'entraînera aucune interruption de service pour les clients connectés. Cependant, la propagation des modifications prendra un certain temps. Pour déterminer à quel moment les modifications se sont complètement propagées pour un cluster Valkey

ou Redis OSS, surveillez la sortie de. `cluster slots` Une fois que tous les nœuds renvoyés par la commande cluster slots ont pris IPs connaissance du nouveau protocole, les modifications ont fini de se propager.

Exemple avec Redis-Py :

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
 nodes):
    nodes = set()

   # This refreshes the cluster topology and will discovery any node updates.
   # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

    time.sleep(1)
```

Exemple avec Lettuce :

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
 6379));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
 switching to IPv4

Set<String> nodes;

do {
   // Check for any changes in the cluster topology.
   // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
    Set<String> nodes = new HashSet<>();

    for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
        nodes.add(node.getUri().getHost());
    }
```

```
    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
            try {
                return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
            } catch (UnknownHostException ignored) {}
            return false;
}));
```

Bonnes pratiques pour les clients (Memcached)

Configuration de votre ElastiCache client pour un équilibrage de charge efficace (Memcached)

> **Note**
>
> Cette section s'applique aux clusters Memcached à plusieurs nœuds auto-conçus.

Pour utiliser efficacement plusieurs nœuds ElastiCache Memcached, vous devez être en mesure de répartir vos clés de cache entre les nœuds. Un moyen simple d'équilibrer la charge d'un cluster à n nœuds consiste à calculer le hachage de la clé de l'objet et à modifier le résultat par n :*hash(key) mod n*. La valeur résultante (0 à n–1) est le numéro du nœud où vous placez l'objet.

Cette approche est simple et fonctionne bien tant que le nombre de nœuds (n) est constant. Cependant, chaque fois que vous ajoutez ou supprimez un nœud du cluster, le nombre de clés qui doivent être déplacées est (n - 1) / n (où n correspond au nouveau nombre de nœuds). Ainsi, cette approche génère le déplacement d'un grand nombre de clés, ce qui se traduit par un grand nombre d'échecs de l'accès aux données du cache initial, surtout quand le nombre de nœuds devient important. La mise à l'échelle de 1 à 2 nœuds entraîne le déplacement de (2-1) / 2 (50 %) des clés, dans le meilleur des cas. La mise à l'échelle de 9 à 10 nœuds entraîne le déplacement de (10-1) / 10 (90 %) des clés. Si vous augmentez à cause d'un pic de trafic, vous ne voulez pas avoir un nombre important d'échecs de l'accès aux données du cache. Un grand nombre d'échecs de l'accès aux données du cache génère des requêtes dans la base de données, qui est déjà surchargée en raison du pic de trafic.

Un hachage cohérent est la solution à ce problème. Un hachage cohérent utilise un algorithme de façon à ce que chaque fois qu'un nœud est ajouté ou supprimé d'un cluster, le nombre de clés qui doivent être déplacées est d'environ 1 / n (où n correspond au nouveau nombre de nœuds). Passer

de 1 à 2 nœuds revient à ce que 1/2 (50 %) des clés soient déplacées, le pire des cas. Passer de 9 à 10 nœuds revient à ce que (10-1) / 10 (10 %) des clés soient déplacées.

En tant que l'utilisateur, vous contrôlez les l'algorithme de hachage qui est utilisé pour les clusters à plusieurs nœuds. Nous vous recommandons de configurer vos clients pour appliquer un hachage cohérent. Heureusement, il existe de nombreuses bibliothèques client Memcached dans les langages les plus utilisés qui mettent en œuvre un hachage cohérent. Consultez la documentation de la bibliothèque que vous utilisez pour voir si elle prend en charge le hachage cohérent et comment le mettre en œuvre.

Si vous travaillez en Java, PHP ou .NET, nous vous recommandons d'utiliser l'une des bibliothèques ElastiCache clientes Amazon.

Hachage cohérent utilisant la technologie Java

Le client Java ElastiCache Memcached est basé sur le client Java open-source spymemcached, qui intègre des fonctionnalités de hachage cohérentes. La bibliothèque inclut une KetamaConnectionFactory classe qui implémente un hachage cohérent. Par défaut, le hachage cohérent est désactivé dans spymemcached.

Pour plus d'informations, consultez la KetamaConnectionFactory documentation à l'adresse KetamaConnectionFactory.

Hachage cohérent à l'aide de PHP avec Memcached

Le client PHP ElastiCache Memcached est un wrapper autour de la bibliothèque PHP Memcached intégrée. Par défaut, la bibliothèque PHP Memcached désactive le hachage cohérent.

Utilisez le code suivant pour activer le hachage cohérent.

```
$m = new Memcached();
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

En plus du code précédent, nous vous conseillons d'activer également `memcached.sess_consistent_hash` dans votre fichier php.ini.

Pour plus d'informations, consultez la documentation de configuration d'exécution pour Memcached PHP à l'adresse http://php. net/manual/en/memcached.configuration.php. Notez spécifiquement le paramètre `memcached.sess_consistent_hash`.

Hachage cohérent à l'aide de .NET avec Memcached

Le client .NET ElastiCache Memcached est un wrapper autour d'Enyim Memcached. Par défaut, le hachage cohérent est activé par le client Enyim Memcached.

Pour plus d'informations, consultez la `memcached/locator` documentation sur https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration# user-content-memcachedlocator.

Clients validés avec Memcached

Les clients suivants ont été spécifiquement validés pour fonctionner avec toutes les configurations de type de réseau prises en charge pour Memcached.

Clients validés :

- AWS ElastiCache Client de cluster Memcached pour Php — Version *3.6.2
- AWS ElastiCache Cluster Client Memcached pour Java — Le dernier master sur Github

Configuration d'un protocole préféré pour les clusters à double pile (Memcached)

Pour les clusters Memcached, vous pouvez contrôler le protocole que les clients utiliseront pour se connecter aux nœuds du cluster à l'aide du paramètre de découverte d'adresses IP. Le paramètre IP Discovery peut être défini sur IPv4 ou IPv6.

Le paramètre de découverte d'adresses IP contrôle le protocole IP utilisé dans la sortie de cluster config get. Ce qui déterminera à son tour le protocole IP utilisé par les clients qui prennent en charge la découverte automatique ElastiCache pour les clusters Memcached.

La modification de la découverte d'adresses IP n'entraînera aucune interruption de service pour les clients connectés. Cependant, la propagation des modifications prendra un certain temps.

Surveillez la sortie de `getAvailableNodeEndPoints` pour Java, et pour Php, surveillez la sortie de `getServerList`. Une fois que la sortie de ces fonctions indique tous IPs les nœuds du cluster qui utilisent le protocole mis à jour, les modifications ont fini de se propager.

Exemple Java :

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
 switching to IPv4
```

```
Set<String> nodes;

do {
    nodes =
 client.getAvailableNodeEndPoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
            try {
                return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
            } catch (UnknownHostException ignored) {}
            return false;
        }));
```

Exemple Php :

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
 expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
    $target_ips_count = 0;

    // For IPv4 use FILTER_FLAG_IPV4
    $target_ips_count = count(array_filter($nodes, function($node) { return
 filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));

} while (count($nodes) !== $target_ips_count);
```

Toutes les connexions client existantes qui ont été créées avant la mise à jour de la découverte d'adresses IP seront toujours connectées à l'aide de l'ancien protocole. Tous les clients validés se reconnecteront automatiquement au cluster à l'aide du nouveau protocole IP une fois que les

modifications seront détectées dans la sortie des commandes de découverte du cluster. Cependant, cela dépend de la mise en œuvre du client.

Clusters à double pile ElastiCache compatibles TLS

Lorsque TLS est activé pour les ElastiCache clusters, les fonctions de découverte de clusters (`cluster slots`,`cluster shards`, et `cluster nodes` pour Redis) ou `config get cluster` pour Memcached renvoient des noms d'hôte au lieu de. IPs Les noms d'hôtes sont ensuite utilisés au lieu de se IPs connecter au ElastiCache cluster et d'effectuer une poignée de main TLS. Cela signifie que les clients ne seront pas affectés par le paramètre de découverte d'adresses IP. Pour les clusters prenant en charge TLS, le paramètre de découverte d'adresses IP n'a aucun effet sur le protocole IP préféré. Au lieu de cela, le protocole IP utilisé sera déterminé par le protocole IP que le client préfère lors de la résolution des noms d'hôtes DNS.

Clients Java

Lorsque vous vous connectez à partir d'un environnement Java qui prend en charge IPv4 les deux IPv6, Java IPv4 privilégiera par défaut IPv6 la rétrocompatibilité. Toutefois, les arguments JVM permettent de configurer la préférence de protocole IP. Pour préférer IPv4, la JVM accepte `-Djava.net.preferIPv4Stack=true` et pour préférer, IPv6 définir`-Djava.net.preferIPv6Stack=true`. Le réglage `-Djava.net.preferIPv4Stack=true` signifie que la JVM n'établira plus aucune IPv6 connexion. Pour Valkey ou Redis OSS, cela inclut ceux destinés à d'autres applications non-Valkey et non-Redis OSS.

Préférences au niveau de l'hôte

En général, si le client ou l'environnement d'exécution du client ne fournit pas d'options de configuration permettant de définir une préférence de protocole IP, lors de la résolution DNS, le protocole IP dépendra de la configuration de l'hôte. Par défaut, la plupart des hôtes IPv6 préfèrent, IPv4 mais cette préférence peut être configurée au niveau de l'hôte. Cela affectera toutes les demandes DNS de cet hôte, et pas seulement celles adressées aux ElastiCache clusters.

Hôtes Linux

Pour Linux, une préférence de protocole IP peut être configurée en modifiant le fichier `gai.conf`. Le fichier `gai.conf` se trouve sous `/etc/gai.conf`. Si `gai.conf` n'est pas spécifié, un exemple doit être disponible sous `/usr/share/doc/glibc-common-x.xx/gai.conf`. Vous pouvez le copier sur `/etc/gai.conf` et la configuration par défaut doit être sans commentaire. Pour mettre à jour la configuration à privilégier IPv4 lors de la connexion à un ElastiCache cluster, mettez à jour la priorité de la plage d'adresses CIDR englobant le cluster IPs afin qu'elle soit supérieure à la priorité

des connexions par défaut. IPv6 Par défaut, IPv6 les connexions ont une priorité de 40. Par exemple, en supposant que le cluster se trouve dans un sous-réseau dont le CIDR est 172.31.0. 0:0 /16, la configuration ci-dessous incitera les clients à préférer les connexions à ce cluster. IPv4

```
label ::1/128       0
label ::/0          1
label 2002::/16     2
label ::/96         3
label ::ffff:0:0/96 4
label fec0::/10     5
label fc00::/7      6
label 2001:0::/32   7
label ::ffff:172.31.0.0/112 8
#
#    This default differs from the tables given in RFC 3484 by handling
#    (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
#    The reason for this difference is that these addresses are never
#    NATed while IPv4 site-local addresses most probably are.  Given
#    the precedence of IPv6 over IPv4 (see below) on machines having only
#    site-local IPv4 and IPv6 addresses a lookup for a global address would
#    see the IPv6 be preferred.  The result is a long delay because the
#    site-local IPv6 addresses cannot be used while the IPv4 address is
#    (at least for the foreseeable future) NATed.  We also treat Teredo
#    tunnels special.
#
# precedence  <mask>   <value>
#    Add another rule to the RFC 3484 precedence table.  See section 2.1
#    and 10.3 in RFC 3484.  The default is:
#
precedence  ::1/128       50
precedence  ::/0          40
precedence  2002::/16     30
precedence ::/96          20
precedence ::ffff:0:0/96  10
precedence ::ffff:172.31.0.0/112 100
```

Plus de détails `gai.conf` sont disponibles sur la page de manuel Linux

Hôtes Windows

Le processus pour les hôtes Windows est similaire. Pour les hôtes Windows, vous pouvez exécuter `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`. Cela a le même effet que la modification du fichier `gai.conf` sur les hôtes Linux.

Cela mettra à jour les politiques de préférence afin de préférer IPv4 les connexions aux IPv6 connexions pour la plage d'adresses CIDR spécifiée. Par exemple, en supposant que le cluster se trouve dans un sous-réseau avec le CIDR 172.31.0. 0:0 /16 en cours d'exécution, le tableau de priorité suivant `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` s'affichera, ce qui incitera les clients à préférer se connecter au cluster. IPv4

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...

Precedence Label Prefix
---------- ----- --------------------------------
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

Gestion de la mémoire réservée pour Valkey et Redis OSS

La mémoire réservée est mise de côté pour être utilisée à des fins autres que le traitement des données. Lors d'une sauvegarde ou d'un basculement, Valkey et Redis OSS utilisent la mémoire disponible pour enregistrer les opérations d'écriture sur votre cluster pendant que les données du cluster sont écrites dans le fichier .rdb. Si vous ne disposez pas de suffisamment de mémoire pour toutes les écritures, le processus échouera. Vous trouverez ci-dessous des informations sur les options de gestion de la mémoire réservée ElastiCache pour Redis OSS et sur la façon d'appliquer ces options.

Rubriques

- De quelle quantité de mémoire réservée avez-vous besoin ?
- Paramètres de gestion de la mémoire réservée
- Spécification de votre paramètre de gestion de la mémoire réservée

De quelle quantité de mémoire réservée avez-vous besoin ?

Si vous utilisez une version de Redis OSS antérieure à la version 2.8.22, réservez plus de mémoire pour les sauvegardes et les basculements que si vous exécutez Redis OSS 2.8.22 ou version ultérieure. Cette exigence est due aux différentes manières dont ElastiCache Redis OSS implémente le processus de sauvegarde. La règle générale est de réserver la moitié de la `maxmemory` valeur d'un type de nœud pour la surcharge de Redis OSS pour les versions antérieures à 2.8.22, et un quart pour les versions 2.8.22 et ultérieures de Redis OSS.

En raison des différentes manières de ElastiCache mettre en œuvre le processus de sauvegarde et de réplication, la règle générale consiste à réserver 25 % de la `maxmemory` valeur d'un type de nœud en utilisant le `reserved-memory-percent` paramètre. Il s'agit de la valeur par défaut recommandée dans la plupart des cas.

Lorsque les microinstances et les petites instances burstables fonctionnent à un niveau proche de leurs `maxmemory` limites, il est possible qu'ils soient soumis à une utilisation du swap. Pour améliorer la fiabilité opérationnelle de ces types d'instances lors de la sauvegarde, de la réplication et d'un trafic élevé, nous recommandons d'augmenter la valeur du `reserved-memory-percent` paramètre jusqu'à 30 % pour les types d'instances de petite taille et jusqu'à 50 % pour les types de micro-instances.

Pour les charges de travail intensives en écriture sur des ElastiCache clusters avec hiérarchisation des données, nous recommandons d'augmenter jusqu'`reserved-memory-percent`à 50 % de la mémoire disponible du nœud.

Pour plus d'informations, consultez les ressources suivantes :

- S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS
- Implémentation de la sauvegarde et de la synchronisation
- Hiérarchisation des données ElastiCache

Paramètres de gestion de la mémoire réservée

Depuis le 16 mars 2017, Amazon ElastiCache fournit deux paramètres mutuellement exclusifs pour gérer votre mémoire Valkey ou Redis OSS, et`reserved-memory`. `reserved-memory-percent` Aucun de ces paramètres ne fait partie de la distribution Valkey ou Redis OSS.

Selon le moment où vous êtes devenu ElastiCache client, l'un ou l'autre de ces paramètres est le paramètre de gestion de la mémoire par défaut. Ce paramètre s'applique lorsque vous créez un nouveau cluster ou groupe de réplication Valkey ou Redis OSS et que vous utilisez un groupe de paramètres par défaut.

- Pour les clients qui ont commencé avant le 16 mars 2017 : lorsque vous créez un cluster Redis OSS ou un groupe de réplication à l'aide du groupe de paramètres par défaut, votre paramètre de gestion de la mémoire est `reserved-memory` Dans ce cas, zéro (0) octets de mémoire sont réservés.
- Pour les clients qui ont commencé le 16 mars 2017 ou après cette date : lorsque vous créez un cluster ou un groupe de réplication Valkey ou Redis OSS à l'aide du groupe de paramètres par défaut, votre paramètre de gestion de la mémoire est `reserved-memory-percent` Dans ce cas, 25 % de la valeur `maxmemory` de votre nœud est réservée à des fins non liées aux données.

Après avoir pris connaissance des deux paramètres de gestion de la mémoire Valkey ou Redis OSS, vous préférerez peut-être utiliser celui qui n'est pas votre paramètre par défaut ou avec des valeurs autres que celles par défaut. Dans ce cas, vous pouvez passer à l'autre paramètre de gestion de la mémoire réservée.

Pour modifier la valeur de ce paramètre, vous pouvez créer un groupe de paramètres personnalisé et le modifier pour utiliser le paramètre et la valeur de gestion de la mémoire préférés. Vous

pouvez ensuite utiliser le groupe de paramètres personnalisé chaque fois que vous créez un nouveau cluster Valkey ou Redis OSS ou un nouveau groupe de réplication. Pour les clusters ou les groupes de réplication existants, vous pouvez les modifier afin d'utiliser votre groupe de paramètres personnalisés.

Pour plus d'informations, veuillez consulter les ressources suivantes :

- Spécification de votre paramètre de gestion de la mémoire réservée
- Création d'un groupe ElastiCache de paramètres
- Modification d'un groupe ElastiCache de paramètres
- Modification d'un ElastiCache cluster
- Modification d'un groupe de réplication

Paramètre reserved-memory

Avant le 16 mars 2017, toute la gestion de ElastiCache la mémoire réservée à Redis OSS était effectuée à l'aide du paramètre`reserved-memory`. La valeur par défaut de `reserved-memory` est 0. Cette valeur par défaut ne réserve aucune mémoire pour la surcharge de Valkey ou Redis OSS et permet à Valkey ou Redis OSS de consommer toute la mémoire d'un nœud en données.

Pour pouvoir modifier `reserved-memory` afin de disposer d'une quantité de mémoire suffisante pour les sauvegardes et les basculements, vous devez créer un groupe de paramètres personnalisé. Dans ce groupe de paramètres personnalisés, vous définissez une valeur adaptée `reserved-memory` à la version de Valkey ou Redis OSS exécutée sur votre cluster et au type de nœud du cluster. Pour de plus amples informations, consultez De quelle quantité de mémoire réservée avez-vous besoin ?.

Le paramètre `reserved-memory` est spécifique à la distribution générale de Redis OSS ElastiCache et ne fait pas partie de celle-ci.

La procédure suivante montre comment gérer la mémoire `reserved-memory` de votre cluster Valkey ou Redis OSS.

Pour réserver de la mémoire à l'aide de la mémoire réservée

1. Créez un groupe de paramètres personnalisé spécifiant la famille de groupe de paramètres correspondant à la version de moteur exécutée par exemple, la famille de groupe de paramètres `redis2.8`. Pour de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

```
aws elasticache create-cache-parameter-group \
   --cache-parameter-group-name redis6x-m3xl \
   --description "Redis OSS 2.8.x for m3.xlarge node type" \
   --cache-parameter-group-family redis6.x
```

2. Calculez le nombre d'octets de mémoire à réserver pour la surcharge de Valkey ou Redis OSS. Vous pouvez trouver la valeur de `maxmemory` du type de nœud dans Paramètres spécifiques au type de nœud Redis OSS.

3. Modifiez le groupe de paramètres personnalisé de façon à ce que le paramètre `reserved-memory` corresponde au nombre d'octets calculé au cours de la première étape. L' AWS CLI exemple suivant suppose que vous exécutez une version de Redis OSS antérieure à la version 2.8.22 et que vous devez réserver la moitié de celle du nœud. `maxmemory` Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis28-m3xl \
   --parameter-name-values "ParameterName=reserved-memory,
 ParameterValue=7130316800"
```

   Notez que vous avez besoin d'un groupe de paramètres personnalisé distinct pour chaque nœud utilisé dans la mesure où chaque type de nœud a une valeur `maxmemory` différente. Par conséquent, chaque type de nœud a besoin d'une valeur différente pour `reserved-memory`.

4. Modifiez votre cluster Redis OSS ou votre groupe de réplication pour utiliser votre groupe de paramètres personnalisé.

   L'exemple de la CLI suivant modifie le cluster `my-redis-cluster` afin qu'il utilise le groupe de paramètres personnalisé `redis28-m3xl` immédiatement. Pour de plus amples informations, veuillez consulter Modification d'un ElastiCache cluster.

```
aws elasticache modify-cache-cluster \
   --cache-cluster-id my-redis-cluster \
   --cache-parameter-group-name redis28-m3xl \
   --apply-immediately
```

   L'exemple de la CLI suivant modifie le groupe de réplication `my-redis-repl-grp` afin qu'il utilise le groupe de paramètres personnalisé `redis28-m3xl` immédiatement. Pour plus d'informations, consultez Modification d'un groupe de réplication.

```
aws elasticache modify-replication-group \
   --replication-group-id my-redis-repl-grp \
   --cache-parameter-group-name redis28-m3xl \
   --apply-immediately
```

Paramètre reserved-memory-percent

Le 16 mars 2017, Amazon ElastiCache a introduit le paramètre `reserved-memory-percent` et l'a rendu disponible sur toutes les versions de ElastiCache pour Redis OSS. L'objectif de `reserved-memory-percent` est de simplifier la gestion de la mémoire réservée sur tous vos clusters. Vous avez ainsi la possibilité d'avoir un seul groupe de paramètres pour chaque famille de groupe de paramètres (par exemple, `redis2.8`) afin de gérer la mémoire réservée de votre cluster, quel que soit le type de nœud. La valeur par défaut de `reserved-memory-percent` est 25 (25 %).

Le paramètre `reserved-memory-percent` est spécifique à la distribution générale de Redis OSS ElastiCache et ne fait pas partie de celle-ci.

Si votre cluster utilise un type de nœud de la famille r6gd et que votre utilisation de la mémoire atteint 75 %, la hiérarchisation des données sera automatiquement déclenchée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

Pour réserver de la mémoire en utilisant reserved-memory-percent

`reserved-memory-percent`Pour gérer la mémoire de votre ElastiCache cluster Redis OSS, effectuez l'une des opérations suivantes :

- Si vous utilisez Redis OSS 2.8.22 ou version ultérieure, attribuez le groupe de paramètres par défaut à votre cluster. Les 25 % par défaut devraient convenir. Si tel n'est pas le cas, procédez comme suit pour modifier la valeur.
- Si vous utilisez une version de Redis OSS antérieure à la version 2.8.22, vous devrez probablement réserver plus de mémoire que la valeur par défaut `reserved-memory-percent` de 25 %. Pour cela, procédez comme suit.

Pour modifier la valeur en pourcentage de reserved-memory-percent

1. Créez un groupe de paramètres personnalisé spécifiant la famille de groupe de paramètres correspondant à la version de moteur exécutée par exemple, la famille de groupe de paramètres

`redis2.8`. Un groupe de paramètres personnalisé est nécessaire car vous ne pouvez pas modifier un groupe de paramètres par défaut. Pour de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

```
aws elasticache create-cache-parameter-group \
   --cache-parameter-group-name redis28-50 \
   --description "Redis OSS 2.8.x 50% reserved" \
   --cache-parameter-group-family redis2.8
```

Dans la mesure où `reserved-memory-percent` réserve de la mémoire sous forme de pourcentage du paramètre `maxmemory` d'un nœud, vous n'avez pas besoin d'un groupe de paramètres personnalisé pour chaque type de nœud.

2. Modifiez le groupe de paramètres personnalisé afin que le paramètre `reserved-memory-percent` soit défini sur 50 (50 %). Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis28-50 \
   --parameter-name-values "ParameterName=reserved-memory-percent,
 ParameterValue=50"
```

3. Utilisez ce groupe de paramètres personnalisé pour tous les clusters ou groupes de réplication Redis OSS exécutant une version de Redis OSS antérieure à 2.8.22.

   L'exemple de CLI suivant modifie le cluster Redis OSS `my-redis-cluster` pour utiliser le groupe de paramètres personnalisés `redis28-50` commençant immédiatement. Pour de plus amples informations, veuillez consulter Modification d'un ElastiCache cluster.

```
aws elasticache modify-cache-cluster \
   --cache-cluster-id my-redis-cluster \
   --cache-parameter-group-name redis28-50 \
   --apply-immediately
```

   L'exemple de CLI suivant modifie le groupe de réplication Redis OSS `my-redis-repl-grp` pour utiliser le groupe de paramètres personnalisés `redis28-50` commençant immédiatement. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.

```
aws elasticache modify-replication-group \
   --replication-group-id my-redis-repl-grp \
```

```
    --cache-parameter-group-name redis28-50 \
    --apply-immediately
```

Spécification de votre paramètre de gestion de la mémoire réservée

Si vous étiez un ElastiCache client actuel le 16 mars 2017, votre paramètre de gestion de mémoire réservée par défaut est `reserved-memory` de zéro (0) octet de mémoire réservée. Si vous êtes devenu ElastiCache client après le 16 mars 2017, votre paramètre de gestion de la mémoire réservée par défaut est `reserved-memory-percent` de réserver 25 % de la mémoire du nœud. Cela est vrai quelle que soit la date à laquelle vous avez créé votre cluster ou groupe de réplication ElastiCache pour Redis OSS. Cependant, vous pouvez modifier le paramètre de gestion de la mémoire réservée à l'aide de l' ElastiCache API AWS CLI or.

Les paramètres `reserved-memory` et `reserved-memory-percent` sont mutuellement exclusifs. Un groupe de paramètres est toujours associé à un paramètre, mais jamais aux deux. Vous pouvez modifier le paramètre utilisé par un groupe de paramètres pour la gestion de la mémoire réservée en modifiant le groupe de paramètres. Le groupe de paramètres doit être un groupe de paramètres personnalisé, car vous ne pouvez pas modifier les groupes de paramètres par défaut. Pour de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

Pour spécifier reserved-memory-percent

Pour utiliser `reserved-memory-percent` comme paramètre de gestion de la mémoire réservée, modifiez un groupe de paramètres personnalisés à l'aide de la commande `modify-cache-parameter-group`. Utilisez le paramètre `parameter-name-values` pour spécifier `reserved-memory-percent` et une valeur pour cela.

L'exemple de la CLI suivant modifie le groupe de paramètres personnalisé `redis32-cluster-on` afin qu'il utilise `reserved-memory-percent` pour gérer la mémoire réservée. Une valeur doit être affectée à `ParameterValue` pour que le groupe de paramètres utilise le paramètre `ParameterName` pour la gestion de la mémoire réservée. Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis32-cluster-on \
   --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

Pour spécifier reserved-memory

Pour utiliser `reserved-memory` comme paramètre de gestion de la mémoire réservée, modifiez un groupe de paramètres personnalisés à l'aide de la commande `modify-cache-parameter-group`. Utilisez le paramètre `parameter-name-values` pour spécifier `reserved-memory` et une valeur pour cela.

L'exemple de la CLI suivant modifie le groupe de paramètres personnalisé `redis32-m3xl` afin qu'il utilise `reserved-memory` pour gérer la mémoire réservée. Une valeur doit être affectée à `ParameterValue` pour que le groupe de paramètres utilise le paramètre `ParameterName` pour la gestion de la mémoire réservée. Comme la version du moteur est plus récente que la version 2.8.22, nous définissons la valeur sur `3565158400`, soit 25 % de `maxmemory` du `cache.m3.xlarge`. Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

```
aws elasticache modify-cache-parameter-group \
   --cache-parameter-group-name redis32-m3xl \
   --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

Meilleures pratiques lors de l'utilisation de clusters conçus par Valkey et Redis OSS

L'utilisation de plusieurs AZ, le fait de disposer d'une mémoire suffisante, le redimensionnement des clusters et la minimisation des temps d'arrêt sont autant de concepts utiles à garder à l'esprit lorsque vous travaillez avec des clusters conçus par vos soins dans Valkey ou Redis OSS. Nous vous recommandons de consulter et de suivre les bonnes pratiques suivantes.

Rubriques

- Réduction des temps d'arrêt avec Multi-AZ
- S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS
- Redimensionnement de cluster en ligne
- Réduction au maximum des temps d'arrêt pendant la maintenance

Réduction des temps d'arrêt avec Multi-AZ

Dans un certain nombre de cas, ElastiCache Valkey ou Redis OSS peuvent 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.

Ce remplacement entraîne un certain temps d'arrêt pour le cluster, mais si Multi-AZ est activé, le temps d'arrêt est réduit. Le rôle du nœud primaire bascule automatiquement sur l'un des réplicas en lecture. Il n'est pas nécessaire de créer et de provisionner un nouveau nœud principal, car il ElastiCache 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.

ConsultezMinimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS, pour en savoir plus sur le multi-AZ et la réduction des temps d'arrêt.

S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS

Instantanés et synchronisations dans Valkey 7.2 et versions ultérieures, et Redis OSS version 2.8.22 et versions ultérieures

Valkey prend en charge par défaut les instantanés et les synchronisations. Redis OSS 2.8.22 introduit un processus de sauvegarde sans fourche qui vous permet d'allouer une plus grande partie de votre mémoire à l'utilisation de votre application sans augmenter l'utilisation des swap lors des synchronisations et des sauvegardes. Pour de plus amples informations, veuillez consulter Implémentation de la sauvegarde et de la synchronisation.

Instantanés et synchronisations Redis OSS antérieurs à la version 2.8.22

Lorsque vous travaillez avec ElastiCache Redis OSS, Redis OSS appelle une commande d'écriture en arrière-plan dans un certain nombre de cas :

- Lorsque de la création d'un instantané pour une sauvegarde.
- Lors de la synchronisation de réplicas avec le réplica principal dans un groupe de réplication.
- Lorsque vous activez la fonctionnalité d'ajout de fichier uniquement (AOF) pour Redis OSS.
- Lors la promotion d'un réplica en tant que maître (qui entraîne une synchronisation du réplica principal/réplica).

Chaque fois que Redis OSS exécute un processus d'écriture en arrière-plan, vous devez disposer de suffisamment de mémoire disponible pour faire face à la surcharge du processus. Si vous ne disposez pas de suffisamment de mémoire, le processus échoue. Pour cette raison, il est important de choisir un type d'instance de nœud disposant de suffisamment de mémoire lors de la création de votre cluster Redis OSS.

Processus d'écriture en arrière-plan et utilisation de la mémoire avec Valkey et Redis OSS

Chaque fois qu'un processus d'écriture en arrière-plan est appelé, Valkey et Redis OSS modifient son processus (n'oubliez pas que ces moteurs sont à thread unique). Un fork conserve vos données sur le disque dans un fichier instantané Redis OSS .rdb. L'autre fork traite toutes les opérations de lecture et d'écriture. Pour garantir que votre instantané est un point-in-time instantané, toutes les mises à jour et tous les ajouts de données sont écrits dans une zone de mémoire disponible distincte de la zone de données.

Tant que vous disposez de suffisamment de mémoire pour enregistrer toutes les opérations d'écriture pendant le stockage des données sur le disque, vous n'aurez pas de problème de mémoire insuffisante. Vous risquez d'avoir des problèmes de mémoire insuffisante si l'une des affirmations suivantes est vraie :

- Votre application effectue de nombreuses opérations d'écriture, nécessitant une grande quantité de mémoire disponible pour accepter les données mises à jour ou nouvelles.
- Vous disposez de très peu de mémoire disponible pour pouvoir écrire de nouvelles données ou mettre à jour des données.
- Stocker durablement sur le disque votre jeu de données volumineux prend du temps car cela nécessite un grand nombre d'opérations d'écriture.

Le schéma suivant représente l'utilisation de la mémoire lors de l'exécution d'un processus d'écriture en arrière-plan.

Pour obtenir des informations sur l'impact d'une sauvegarde sur les performances, consultez Impact sur les performances des sauvegardes de clusters auto-conçus.

Pour plus d'informations sur la façon dont Valkey et Redis OSS exécutent les snapshots, consultez http://valkey.io.

Pour plus d'informations sur les régions et les zones de disponibilité, consultez Choix des régions et des zones de disponibilité pour ElastiCache.

Eviter tout dépassement de mémoire lors de l'exécution d'un processus d'écriture en arrière-plan

Chaque fois qu'un processus d'écriture en arrière-plan tel que `BGSAVE` ou `BGREWRITEAOF` est appelé, pour éviter l'échec du processus, vous devez disposer de plus de mémoire disponible que celle qui sera consommée par les opérations d'écriture pendant le processus. Dans le pire des cas, pendant l'opération d'écriture en arrière-plan, chaque enregistrement est mis à jour et de nouveaux enregistrements sont ajoutés au cache. `reserved-memory-percent`Pour cette raison, nous vous recommandons de définir la valeur 50 (50 %) pour les versions de Redis OSS antérieures à la version 2.8.22, ou 25 (25 %) pour Valkey et toutes les versions de Redis OSS 2.8.22 et ultérieures.

La valeur `maxmemory` indique que la mémoire dont vous disposez pour le traitement des données et la surcharge opérationnelle. Etant donné que vous ne pouvez pas modifier le paramètre `reserved-memory` dans le groupe de paramètres par défaut, vous devez créer un groupe de paramètres personnalisés pour le cluster. La valeur par défaut `reserved-memory` est 0, ce qui permet à Redis OSS de consommer toute la mémoire maximale associée aux données, laissant potentiellement trop peu de mémoire pour d'autres utilisations, telles qu'un processus d'écriture en arrière-plan. Pour les valeurs `maxmemory` par type d'instance de nœud, consultez Paramètres spécifiques au type de nœud Redis OSS.

Vous pouvez également utiliser le `reserved-memory` paramètre pour réduire la quantité de mémoire utilisée sur le boîtier.

Pour plus d'informations sur les paramètres spécifiques à Valkey et Redis dans, consultez. ElastiCache Paramètres OSS Valkey et Redis

Pour plus d'informations sur la création et la modification des groupes de paramètres, consultez Création d'un groupe ElastiCache de paramètres et Modification d'un groupe ElastiCache de paramètres.

Redimensionnement de cluster 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 ElastiCache les métriques Redis OSS 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 `BytesUsedForCacheItems`.
- 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.

- Chargez vos fonctions sur chaque partition : lorsque vous agrandissez votre cluster, les fonctions chargées dans l'un des nœuds existants (sélectionnées au hasard) ElastiCache seront automatiquement répliquées sur le ou les nouveaux nœuds. Si votre cluster possède Valkey 7.2 ou une version ultérieure, ou Redis OSS 7.0 ou une version ultérieure, et que votre application utilise Functions, nous vous recommandons de charger toutes vos fonctions sur toutes les partitions avant de les redimensionner afin que votre cluster ne se retrouve pas avec des fonctions différentes sur différentes partitions.

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. Les données des partitions cibles ne seront pas supprimées.
- 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. Avant Redis OSS 6, la `BRPOPLPUSH` commande n'est pas prise en charge si elle fonctionne sur le slot en cours de migration.

Réduction au maximum des temps d'arrêt pendant la maintenance

La configuration en mode cluster présente la meilleure disponibilité pendant les opérations gérées ou non gérées. Nous vous recommandons d'utiliser un client pris en charge en mode cluster qui se connecte au point de terminaison de détection des clusters. Pour le mode cluster désactivé, nous vous recommandons d'utiliser le point de terminaison principal pour toutes les opérations d'écriture.

Pour l'activité de lecture, les applications peuvent également se connecter à n'importe quel nœud du cluster. Contrairement au point de terminaison principal, les points de terminaison des nœuds sont résolus dans des points de terminaison spécifiques. En cas de modification dans votre cluster, comme l'ajout ou la suppression d'un réplica, vous devez mettre à jour les points de terminaison des nœuds de votre application. C'est la raison pour laquelle, en mode cluster désactivé, nous vous recommandons d'utiliser le point de terminaison du lecteur pour les activités de lecture.

S' AutoFailover il est activé dans le cluster, le nœud principal peut changer. Ainsi, l'application doit confirmer le rôle du nœud et mettre à jour tous les points de terminaison en lecture. Cette étape permet de garantir qu'une charge majeure n'est pas imposée sur le nœud principal. Lorsque

AutoFailover cette option est désactivée, le rôle du nœud ne change pas. Cependant, le temps d'arrêt des opérations gérées ou non gérées est plus élevé que celui des clusters AutoFailover activés.

Évitez de diriger les demandes de lecture vers un seul nœud de réplica en lecture, car son indisponibilité pourrait entraîner une interruption de lecture. Revenez à la lecture depuis le nœud primaire ou assurez-vous de disposer d’au moins deux réplicas en lecture pour éviter toute interruption de lecture pendant la maintenance.

Stratégies de mise en cache pour Memcached

Dans la rubrique suivante, vous trouverez des stratégies pour remplir et gérer votre cache Memcached.

La ou les stratégies que vous souhaitez mettre en œuvre pour remplir et assurer la maintenance de votre cache dépendent des données vous avez mises en cache et des modèles d'accès aux données. Par exemple, vous ne souhaiterez sans doute pas utiliser la même stratégie pour le top 10 d'un jeu sur Internet ou pour des sujets d'actualité. Dans le reste de cette section, nous aborderons les stratégies de maintenance de cache courantes, leurs avantages et leurs inconvénients.

Rubriques

- Réplicas en lecture
- Chargement différé
- Écriture simultanée
- Ajout d'une durée de vie
- Rubriques en relation

Réplicas en lecture

Vous pouvez souvent améliorer de manière significative les performances des caches ElastiCache sans serveur en créant des répliques et en les lisant au lieu du nœud de cache principal. Pour de plus amples informations, veuillez consulter Bonnes pratiques d'utilisation des répliques de lecture.

Chargement différé

Comme le nom l'indique, un chargement différé correspond à une stratégie de mise en cache qui charge des données dans la mémoire cache uniquement lorsque c'est nécessaire. Cela fonctionne comme décrit ci-dessous.

Amazon ElastiCache est un magasin clé-valeur en mémoire situé entre votre application et le magasin de données (base de données) auquel elle accède. Chaque fois que votre application demande des données, elle les envoie d'abord au ElastiCache cache. Si les données existent dans le cache et sont à jour, ElastiCache renvoie les données dans votre application. Si les données n'existent pas dans le cache ou ont expiré, votre application demande les données à partir de votre magasin de données. Votre magasin de données renvoie ensuite les données à votre application. Votre application enregistre les données reçues du magasin dans le cache. De cette façon, elles peuvent être récupérées plus rapidement la prochaine fois qu'elles sont demandées.

Une correspondance avec le cache se produit lorsque les données sont dans le cache et n'ont pas expiré :

1. L'application demande des données dans le cache.
2. Le cache retourne les données à l'application.

Une absence de données dans le cache se produit lorsque les données ne sont pas dans le cache ou ont expiré :

1. L'application demande des données dans le cache.
2. Le cache n'a pas les données demandées, un code `null` est donc retourné.
3. L'application demande et reçoit les données à partir de la base de données.
4. L'application met à jour le cache avec les nouvelles données.

Avantages et inconvénients du chargement différé

Les avantages du chargement différé sont les suivants :

- Seules les données demandées sont mises en cache.

  Dans la mesure où la plupart des données ne sont jamais demandées, le chargement différé évite de remplir le cache avec des données qui ne sont pas nécessaires.
- Les défaillances de nœud ne sont pas fatales pour votre application.

  Lorsqu'un nœud échoue et qu'il est remplacé par un nœud vide, l'application continue à fonctionner, mais avec une latence accrue. Lorsque les requêtes sont effectuées vers le nouveau nœud, chaque échec de cache entraîne une requête dans la base de données. En même temps, la copie de données est ajoutée au cache afin que les requêtes suivantes soient récupérées à partir du cache.

Les inconvénients du chargement différé sont les suivants :

- Il y a une pénalité pour les échecs de cache. Chaque échec de cache génère 3 sorties :
  1. Une demande initiale de données à partir du cache
  2. Une requête pour les données dans la base de données
  3. L'enregistrement des données dans le cache

Ce qui peut générer un retard notable dans l'acheminement des données vers l'application.

- Données obsolètes.

  Si les données sont écrites dans le cas d'un échec de cache, les données du cache deviennent périmées. Ce résultat se produit car il n'y a pas de mise à jour dans le cache quand les données sont modifiées dans la base de données. Pour résoudre ce problème, vous pouvez utiliser les stratégies Écriture simultanée et Ajout d'une durée de vie.

Exemple de pseudocode de chargement différé

Le code suivant est un exemple de pseudo-code de la logique d'un chargement différé.

```
// ****************************************
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//    retrieved from the database,
//    added to the cache, and
//    returned to the application
// ****************************************
get_customer(customer_id)

    customer_record = cache.get(customer_id)
    if (customer_record == null)

        customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",
 customer_id)
        cache.set(customer_id, customer_record)

    return customer_record
```

Dans cet exemple, le code d'application qui obtient les données est le suivant.

```
customer_record = get_customer(12345)
```

Écriture simultanée

La stratégie d'écriture simultanée permet d'ajouter des données et de mettre à jour les données du cache dès que des données sont enregistrées dans la base de données.

Avantages et les inconvénients de l'écriture simultanée

Les avantages de l'écriture simultanée sont les suivants :

- Les données dans le cache ne se périment jamais.

  Dans la mesure où les données dans le cache sont mises à jour à chaque enregistrement dans la base de données, les données dans le cache sont toujours actuelles.
- Pénalité d'écriture vs pénalité de lecture.

  Chaque écriture implique deux sorties :

  1. Une écriture dans le cache
  2. Une écriture dans la base de données

  Ce qui ajoute une latence pour le processus. Cela dit, les utilisateurs finaux acceptent généralement plus facilement la latence lors de l'actualisation que lors de la récupération des données. On sait communément que les mises à jour demandent plus de travail et donc prennent plus de temps.

Les inconvénients de l'écriture simultanée sont les suivants :

- Données manquantes.

  Si vous mettez en service un nouveau nœud, que ce soit en raison d'une panne de nœud ou d'une mise à l'échelle, il y a des données manquantes. Ces données restent manquantes jusqu'à ce qu'elles soient ajoutées ou mises à jour dans la base de données. Vous pouvez minimiser cela en implémentant le chargement différé avec l'écriture simultanée.
- Evolution du cache.

  La plupart des données ne sont jamais lues, ce qui est un gaspillage de ressources. En ajoutant une valeur de time to live (TTL), vous pouvez minimiser l'espace perdu.

Exemple de pseudocode d'écriture simultanée

Le code suivant est un exemple de pseudo-code de logique d'écriture simultanée.

```
// ****************************************
// function that saves a customer's record.
// ****************************************
```

```
save_customer(customer_id, values)

    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
    cache.set(customer_id, customer_record)
    return success
```

Dans cet exemple, le code d'application qui obtient les données est le suivant.

```
save_customer(12345,{"address":"123 Main"})
```

Ajout d'une durée de vie

Le chargement différé permet l'existence de données obsolètes, mais n'échouera pas sur les nœuds vides. L'écriture simultanée garantit que les données sont toujours parfaitement à jour, mais peuvent échouer sur des nœuds vides et remplir le cache de données superflues. En ajoutant une time to live (TTL) à chaque écriture, vous pouvez avoir les avantages de chaque stratégie. En même temps, vous pouvez et en grande partie éviter d'encombrer le cache avec des données supplémentaires.

Le temps de vie (TTL) est une valeur entière qui indique le nombre de secondes avant l'expiration de la clé. Valkey ou Redis OSS peuvent spécifier des secondes ou des millisecondes pour cette valeur. Memcached spécifie cette valeur en secondes. Lorsqu'une application tente de lire une clé expirée, c'est comme si la clé était introuvable. La base de données est interrogée pour la clé et le cache est mis à jour. Cette approche ne garantit pas qu'une valeur n'est pas obsolète. Cependant, il empêche les données de devenir trop anciennes et exige que les valeurs du cache soient occasionnellement rafraîchies à partir de la base de données.

Pour plus d'informations, consultez les commandes Valkey et Redis OSS ou les commandes `set` Memcached.

Exemples de pseudocode TTL

Le code suivant est un exemple de pseudo-code de logique d'écriture simultanée comportant un un time to live.

```
// ****************************************
// function that saves a customer's record.
// The TTL value of 300 means that the record expires
//    300 seconds (5 minutes) after the set command
//    and future reads will have to query the database.
// ****************************************
save_customer(customer_id, values)
```

```
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)
    cache.set(customer_id, customer_record, 300)

    return success
```

Le code suivant est un exemple de pseudo-code de logique de chargement différé comportant un time to live.

```
// ****************************************
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//    retrieved from the database,
//    added to the cache, and
//    returned to the application.
// The TTL value of 300 means that the record expires
//    300 seconds (5 minutes) after the set command
//    and subsequent reads will have to query the database.
// ****************************************
get_customer(customer_id)

    customer_record = cache.get(customer_id)

    if (customer_record != null)
        if (customer_record.TTL < 300)
            return customer_record        // return the record and exit function

    // do this only if the record did not exist in the cache OR
    //    the TTL was >= 300, i.e., the record in the cache had expired.
    customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
    cache.set(customer_id, customer_record, 300)  // update the cache
    return customer_record                // return the newly retrieved record and exit
 function
```

Dans cet exemple, le code d'application qui obtient les données est le suivant.

```
save_customer(12345,{"address":"123 Main"})
```

```
customer_record = get_customer(12345)
```

Rubriques en relation

- Stockage de données en mémoire
- Choix d'un moteur et d'une version
- Dimensionnement ElastiCache

Gestion de votre cluster conçu par vos soins dans ElastiCache

ElastiCache propose deux options de déploiement, la mise en cache sans serveur et les clusters conçus par ses soins. Chacun a ses propres capacités et exigences.

Cette section contient des rubriques destinées à vous aider à gérer les clusters que vous avez conçus vous-même.

> **Note**
>
> Ces rubriques ne s'appliquent pas à ElastiCache Serverless.

Rubriques

- Clusters Auto Scaling Valkey et Redis OSS
- Modification du mode cluster
- Réplication entre AWS régions à l'aide de banques de données mondiales
- Haute disponibilité avec les groupes de réplication
- Gestion de la maintenance des ElastiCache clusters
- Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres

Clusters Auto Scaling Valkey et Redis OSS

Prérequis

ElastiCache Auto Scaling se limite aux éléments suivants :

- Clusters Valkey ou Redis OSS (mode cluster activé) exécutant Valkey 7.2 ou version ultérieure de Redis OSS 6.0

- Hiérarchisation des données (mode cluster activé) clusters exécutant Valkey 7.2 et versions ultérieures ou Redis OSS 7.0.7 et versions ultérieures
- Tailles d'instance : grande XLarge, 2 XLarge
- Familles de types d'instances : R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- Auto Scaling in n' ElastiCache est pas pris en charge pour les clusters exécutés dans des banques de données mondiales, des Outposts ou des Zones Locales.

Gestion automatique de la capacité avec ElastiCache Auto Scaling avec Valkey ou Redis OSS

ElastiCache la mise à l'échelle automatique avec Valkey ou Redis OSS permet d'augmenter ou de diminuer automatiquement les partitions ou les répliques souhaitées dans votre service. ElastiCache ElastiCache utilise le service Application Auto Scaling pour fournir cette fonctionnalité. Pour en savoir plus, veuillez consulter Application Auto Scaling. Pour utiliser le dimensionnement automatique, vous définissez et appliquez une politique de dimensionnement qui utilise CloudWatch les métriques et les valeurs cibles que vous attribuez. ElastiCache Auto Scaling utilise la politique pour augmenter ou diminuer le nombre d'instances en réponse aux charges de travail réelles.

Vous pouvez utiliser le AWS Management Console pour appliquer une politique de dimensionnement basée sur une métrique prédéfinie. Une métrique `predefined metric` est définie dans une énumération de telle sorte que vous pouvez la spécifier par son nom dans le code ou l'utiliser dans la AWS Management Console. Les métriques personnalisées ne sont pas disponibles pour la sélection à l'aide de la AWS Management Console. Vous pouvez également utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling pour appliquer une politique de dimensionnement basée sur une métrique prédéfinie ou personnalisée.

ElastiCache pour Valkey et Redis, OSS prend en charge le dimensionnement pour les dimensions suivantes :

- Partitions – Ajouter/supprime automatiquement des partitions dans le cluster de manière similaire au repartitionnement manuel en ligne. Dans ce cas, le dimensionnement ElastiCache automatique déclenche le dimensionnement en votre nom.
- Répliques — add/remove replicas in the cluster similar to manual Increase/Decrease replica operations. ElastiCache auto scaling for Valkey and Redis OSS adds/removes Réplique automatiquement et de manière uniforme sur toutes les partitions du cluster.

ElastiCache pour Valkey et Redis OSS prend en charge les types de politiques de dimensionnement automatique suivants :

- Politiques de dimensionnement Suivi de la cible— Augmentez ou diminuez le nombre shards/replicas d'exécutions de votre service en fonction d'une valeur cible pour une métrique spécifique. Cette option est similaire à la façon dont votre thermostat maintient la température de votre domicile. Vous sélectionnez une température et le thermostat se charge du reste.
- Dimensionnement planifié pour votre application. — ElastiCache pour Valkey et Redis OSS, le dimensionnement automatique peut augmenter ou diminuer le nombre d'exécutions de shards/replicas votre service en fonction de la date et de l'heure.

Les étapes suivantes résument le processus de mise à l'échelle automatique ElastiCache pour Valkey et Redis OSS, comme indiqué dans le schéma précédent :

1. Vous créez une politique de dimensionnement ElastiCache automatique pour votre groupe de réplication.
2. ElastiCache Auto Scaling crée une paire d' CloudWatch alarmes en votre nom. Chaque paire représente vos limites supérieure et inférieure pour les métriques. Ces CloudWatch alarmes sont déclenchées lorsque l'utilisation réelle du cluster s'écarte de votre utilisation cible pendant une période prolongée. Vous pouvez afficher les alarmes dans la console.
3. Si la valeur de mesure configurée dépasse votre objectif d'utilisation (ou tombe en dessous de l'objectif) pendant une période donnée, CloudWatch déclenche une alarme qui déclenche le dimensionnement automatique pour évaluer votre politique de dimensionnement.
4. ElastiCache auto scaling émet une demande de modification pour ajuster la capacité de votre cluster.
5. ElastiCache traite la demande de modification en augmentant (ou en diminuant) dynamiquement la Shards/Replicas capacité du cluster afin qu'elle se rapproche de votre objectif d'utilisation.

Pour comprendre le fonctionnement d' ElastiCache Auto Scaling, supposons que vous ayez un cluster nommé`UsersCluster`. En surveillant les CloudWatch métriques`UsersCluster`, vous déterminez le nombre maximum de partitions dont le cluster a besoin lorsque le trafic est à son maximum et le nombre minimal de partitions lorsque le trafic est à son point le plus bas. Vous définissez également une valeur cible pour l'utilisation du processeur pour le `UsersCluster` cluster. ElastiCache auto scaling utilise son algorithme de suivi des cibles pour s'assurer que les partitions provisionnées `UsersCluster` sont ajustées selon les besoins afin que l'utilisation reste égale ou proche de la valeur cible.

> **Note**
>
> La mise à l'échelle peut prendre un certain temps et nécessitera des ressources de cluster supplémentaires pour que les partitions puissent être rééquilibrées. ElastiCache Auto Scaling modifie les paramètres des ressources uniquement lorsque la charge de travail réelle reste élevée (ou diminuée) pendant une période prolongée de plusieurs minutes. L'algorithme de suivi des cibles à mise à l'échelle automatique vise à maintenir l'utilisation de la cible à la valeur que vous avez choisie ou à un niveau proche de celle-ci sur le long terme.

Politiques Auto Scaling

Une politique de mise à l'échelle comporte les éléments suivants :

- Une métrique cible : métrique utilisée par Valkey et Redis OSS Auto Scaling ElastiCache pour déterminer à quel moment et dans quelle CloudWatch mesure il convient de procéder à une mise à l'échelle.
- Capacités maximale et minimale – Le nombre minimal et maximal de partitions ou de réplicas à utiliser pour la mise à l'échelle.

> **Important**
>
> Lors de la création de la politique de dimensionnement automatique, si la capacité actuelle est supérieure à la capacité maximale configurée, nous l'adaptons MaxCapacity lors de la création de la politique. De même, si la capacité actuelle est inférieure à la capacité minimale configurée, nous augmentons jusqu'à. MinCapacity

- Un temps de stabilisation – La durée, en secondes, entre la fin d'une activité de mise à l'échelle horizontale montante ou descendante et le début d'une autre activité de mise à l'échelle horizontale.
- Rôle lié à un service : rôle AWS Identity and Access Management (IAM) lié à un service spécifique. AWS Un rôle lié à un service inclut toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom. ElastiCache Auto Scaling génère automatiquement ce `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` rôle pour vous.
- Activer ou désactiver les activités de dimensionnement en charge – Capacité d'activer ou désactiver les activités de dimensionnement en charge pour une politique.

Rubriques

- Métrique cible pour Auto Scaling
- Capacité minimale et maximale
- Temps de stabilisation
- Activation ou désactivation d'activités de diminution en charge

Métrique cible pour Auto Scaling

Dans ce type de stratégie, une métrique prédéfinie ou personnalisée et une valeur cible pour la métrique sont spécifiées dans une configuration de politique de dimensionnement du suivi des cibles. ElastiCache pour Valkey et Redis OSS Auto Scaling crée et gère les CloudWatch alarmes qui déclenchent la politique de dimensionnement et calcule l'ajustement de mise à l'échelle en fonction

de la métrique et de la valeur cible. La politique de dimensionnement ajoute ou supprime shards/replicas selon les besoins pour maintenir la métrique à la valeur cible spécifiée ou proche de celle-ci. En plus de maintenir la métrique proche de la valeur cible, une politique de dimensionnement Suivi de la cible s'ajuste également aux fluctuations de la métrique dues à l'évolution de la charge de travail. Une telle politique minimise également les fluctuations rapides du nombre de produits disponibles shards/replicas pour votre cluster.

Par exemple, examinons une politique de mise à l'échelle qui utilise la métrique prédéfinie d'utilisation moyenne de `ElastiCachePrimaryEngineCPUUtilization`. Ce type de politique peut maintenir l'utilisation du CPU au pourcentage d'utilisation indiqué, tel que 70 %, ou proche de celui-ci.

> **Note**
>
> Pour chaque cluster, vous ne pouvez créer qu'une seule politique Auto Scaling pour chaque métrique cible.

Capacité minimale et maximale

Partitions

Vous pouvez spécifier le nombre maximum de partitions pouvant être redimensionnées ElastiCache pour le dimensionnement automatique de Valkey et Redis OSS. Cette valeur doit être comprise entre 1 et 250, inclus. Vous pouvez également spécifier le nombre minimum de partitions à gérer par le biais de la mise à l'échelle automatique. Cette valeur doit être au moins égale à 1, et égale ou inférieure à la valeur spécifiée pour le maximum de partitions, soit 250.

Réplicas

Vous pouvez spécifier le nombre maximum de répliques à gérer pour le dimensionnement automatique de ElastiCache Valkey et Redis OSS. Cette valeur doit être inférieure ou égale à 5. Vous pouvez également spécifier le nombre minimum de répliques à gérer par le biais du dimensionnement automatique. Cette valeur doit être au moins égale à 1, et égale ou inférieure à la valeur spécifiée pour le maximum de réplicas, soit 5.

Pour déterminer le nombre minimum et maximum shards/replicas dont vous avez besoin pour le trafic type, testez votre configuration Auto Scaling avec le taux de trafic attendu vers votre modèle.

> **Note**
>
> ElastiCache les politiques de dimensionnement automatique augmentent la capacité du cluster jusqu'à ce qu'il atteigne la taille maximale que vous avez définie ou jusqu'à ce que des limites de service s'appliquent. Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

> **Important**
>
> Dimensionnement en l'absence de trafic. Si le trafic d'une variante devient nul, il est ElastiCache automatiquement ajusté au nombre minimum d'instances spécifié.

Temps de stabilisation

Vous pouvez affiner la réactivité d'une politique de suivi de cible et d'échelonnement en ajoutant des temps de stabilisation qui affectent le dimensionnement de votre cluster. Un temps de stabilisation bloque les demandes de montée ou de diminution en charge ultérieures jusqu'à l'expiration de la période. Cela ralentit les suppressions de shards/replicas votre cluster ElastiCache for Valkey et Redis OSS pour les requêtes scale-in, et la création de requêtes scale-out. shards/replicas Vous pouvez spécifier les temps de stabilisation suivants :

- Une activité de scale-in réduit le nombre de membres de votre shards/replicas cluster. Un temps de stabilisation de diminution en charge spécifie la durée, en secondes, devant s'écouler entre la fin d'une activité de diminution et le début d'une autre.
- Une activité de scale-out augmente le nombre de membres de votre shards/replicas cluster. Un temps de stabilisation de montée en charge spécifie la durée, en secondes, devant s'écouler entre la fin d'une activité de montée en charge et le début d'une autre.

Lorsqu'aucun temps de stabilisation de diminution ou de montée en charge n'est pas spécifié, la valeur par défaut est 600 secondes pour l'augmentation et de 900 secondes pour la diminution.

Activation ou désactivation d'activités de diminution en charge

Vous pouvez activer ou désactiver des activités de diminution en charge pour une politique. L'activation des activités d'extension permet de supprimer la politique de dimensionnement. shards/

replicas. When scale-in activities are enabled, the scale-in cooldown period in the scaling policy applies to scale-in activities. Disabling scale-in activities prevents the scaling policy from deleting shards/replicas

> **Note**
>
> Les activités de scale-out sont toujours activées afin que la politique de dimensionnement puisse créer des ElastiCache partitions ou des répliques selon les besoins.

Autorisations IAM requises pour Auto Scaling

ElastiCache pour Valkey et Redis OSS Auto Scaling est rendu possible par une combinaison de ElastiCache CloudWatch, et Application Auto Scaling. APIs Les clusters sont créés et mis à jour avec ElastiCache, les alarmes sont créées avec CloudWatch et les politiques de dimensionnement sont créées avec Application Auto Scaling. Outre les autorisations IAM standard pour créer et mettre à jour des clusters, l'utilisateur IAM qui accède aux paramètres d' ElastiCache Auto Scaling doit disposer des autorisations appropriées pour les services qui prennent en charge le dimensionnement dynamique. Dans cette politique la plus récente, nous avons ajouté la prise en charge de la mise à l'échelle verticale de Memcached, avec l'action. `elasticache:ModifyCacheCluster` Les utilisateurs IAM doivent disposer des autorisations nécessaires pour utiliser les actions indiquées dans l'exemple de politique suivant :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "elasticache:DescribeReplicationGroups",
                "elasticache:ModifyReplicationGroupShardConfiguration",
                "elasticache:IncreaseReplicaCount",
                "elasticache:DecreaseReplicaCount",
                "elasticache:DescribeCacheClusters",
                "elasticache:DescribeCacheParameters",
                "cloudwatch:DeleteAlarms",
```

```
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-
cluster"
        }
    ]
}
```

Rôle lié à un service

Le service de dimensionnement automatique ElastiCache pour Valkey et Redis OSS a également besoin d'une autorisation pour décrire vos clusters et vos CloudWatch alarmes, ainsi que d'autorisations pour modifier votre capacité ElastiCache cible en votre nom. Si vous activez Auto Scaling pour votre cluster, il crée un rôle lié à un service nommé. `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` Ce rôle lié au service accorde à ElastiCache Auto Scaling l'autorisation de décrire les alarmes correspondant à vos politiques, de surveiller la capacité actuelle du parc et de modifier la capacité du parc. Le rôle lié au service est le rôle par défaut pour le dimensionnement ElastiCache automatique. Pour plus d'informations, consultez la section Rôles liés à un service ElastiCache pour Redis OSS Auto Scaling dans le Guide de l'utilisateur d'Application Auto Scaling.

Bonnes pratiques pour Auto Scaling

Avant de vous inscrire à Auto Scaling, nous vous recommandons de procéder comme suit :

1. Utiliser une seule métrique de suivi : identifiez si votre cluster a des charges de travail gourmandes en processeur ou en données et utilisez une métrique prédéfinie correspondante pour définir la politique de mise à l'échelle.

- Processeur du moteur : `ElastiCachePrimaryEngineCPUUtilization` (dimension de la partition) ou `ElastiCacheReplicaEngineCPUUtilization` (dimension du réplica)
- Utilisation de la base de données : `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`. Cette politique de mise à l'échelle fonctionne de manière optimale lorsque maxmemory-policy est définie sur noeviction sur le cluster.

Nous vous recommandons d'éviter d'appliquer plusieurs politiques par dimension sur le cluster. ElastiCache pour Valkey et Redis OSS Auto Scaling élargira la cible évolutive si des politiques de suivi des cibles sont prêtes à être étendues, mais ne sera étendue que si toutes les politiques de suivi des cibles (avec la partie scale-in activée) sont prêtes à être étendues. Si plusieurs politiques indiquent simultanément à la cible évolutive de procéder à une montée en puissance ou à une diminution de charge, elle effectue la mise à l'échelle en fonction de la politique qui fournit la plus grande capacité à la fois pour la montée et la diminution en charge.

2. Métriques personnalisées pour le suivi de cibles : soyez prudent lorsque vous utilisez des métriques personnalisées pour le suivi de cibles, car la fonction Auto Scaling est mieux adaptée à la montée en puissance/la mise à l'échelle horizontale proportionnelle aux changements dans les métriques choisies pour la politique. Si ces métriques ne changent pas de manière proportionnelle pour les actions de mise à l'échelle utilisées pour la création de politiques, cela peut entraîner des actions de montée en puissance et de mise à l'échelle horizontale continues pouvant affecter la disponibilité ou le coût.

   Pour les clusters avec hiérarchisation des données (types d'instances de la famille r6gd), évitez d'utiliser des métriques basées sur la mémoire pour la mise à l'échelle.

3. Dimensionnement planifié : si vous constatez que votre charge de travail est déterministe (high/low atteinte à un moment précis), nous vous recommandons d'utiliser le dimensionnement planifié et de configurer votre capacité cible en fonction des besoins. Le suivi de cible est mieux adapté aux charges de travail non déterministes et au cluster pour fonctionner à la métrique cible requise en augmentant la capacité lorsque vous avez besoin de plus de ressources et en la diminuant lorsque vous en avez besoin de moins.
4. Désactiver la mise à l'échelle interne : le dimensionnement automatique sur Target Tracking convient parfaitement aux clusters dont les charges increase/decrease de travail sont progressives, car les indicateurs peuvent déclencher des oscillations de spikes/dip scale-out/in consécutives. Afin d'éviter de telles oscillations, vous pouvez commencer par désactiver la mise à l'échelle horizontale et, par la suite, vous pouvez toujours adapter manuellement à vos besoins.

5. Testez votre application : nous vous recommandons de tester votre application avec vos Min/Max charges de travail estimées afin de déterminer les valeurs minimales et maximales absolues shards/replicas requises pour le cluster, tout en créant des politiques de dimensionnement pour éviter les problèmes de disponibilité. La scalabilité automatique peut monter en puissance la capacité jusqu'au seuil Max et mettre à l'échelle horizontale la capacité jusqu'au seuil Min configuré pour la cible.
6. Définition de la valeur cible : vous pouvez analyser les CloudWatch mesures correspondantes relatives à l'utilisation du cluster sur une période de quatre semaines afin de déterminer le seuil de valeur cible. Si vous n'êtes toujours pas sûr de la valeur à choisir, nous vous recommandons de commencer par une valeur de métrique prédéfinie minimale prise en charge.
7. AutoScaling sur Target Tracking convient parfaitement aux clusters dotés d'une répartition uniforme des charges de travail entre les shards/replicas dimensions. Une distribution non uniforme peut conduire à :
   - Mise à l'échelle lorsque cela n'est pas nécessaire en raison de spike/dip la charge de travail sur quelques hot shards/répliques.
   - L'absence de mise à l'échelle lorsque cela est nécessaire en raison d'une moyenne globale proche de l'objectif, même si l'on dispose de partitions/réplicas chauds.

> **Note**
>
> Lorsque vous agrandissez votre cluster, les fonctions chargées dans l'un des nœuds existants (sélectionnées au hasard) ElastiCache seront automatiquement répliquées sur le ou les nouveaux nœuds. Si votre cluster utilise Valkey ou Redis OSS 7.0 ou une version ultérieure et que votre application utilise Functions, nous vous recommandons de charger toutes vos fonctions sur toutes les partitions avant de les redimensionner afin que votre cluster ne se retrouve pas avec des fonctions différentes sur différentes partitions.

Après vous être inscrit auprès de AutoScaling, notez ce qui suit :

- Il existe des limitations sur les configurations de scalabilité automatique prises en charge, nous vous recommandons donc de ne pas modifier la configuration d'un groupe de réplication qui est inscrit pour la scalabilité automatique. Voici quelques exemples :
  - Modifier manuellement le type d'instance vers des types non pris en charge.
  - Association du groupe de réplication à un entrepôt de données global.

- Modification du paramètre `ReservedMemoryPercent`.
- increasing/decreasing shards/replicas beyond the Min/MaxCapacité configurée manuellement lors de la création de la politique.

Utilisation de Auto Scaling avec des partitions

Avec ElastiCache's, AutoScaling vous pouvez utiliser les politiques de suivi et de planification avec votre moteur Valkey ou Redis OSS.

Vous trouverez ci-dessous des informations sur le suivi des cibles et les politiques planifiées, ainsi que sur la manière de les appliquer à l'aide du AWS Management Console AWS CLI et APIs.

Rubriques

- Politiques de dimensionnement Suivi de la cible
- Ajout d'une politique de mise à l'échelle
- Enregistrement d'une cible évolutive
- Définition d'une politique de mise à l'échelle
- Désactivation de l'activité de diminution en charge
- Application d'une politique de mise à l'échelle
- Modification d'une politique de dimensionnement
- Suppression d'une politique de dimensionnement
- Utilisation AWS CloudFormation pour les politiques Auto Scaling
- Mise à l'échelle planifiée

Politiques de dimensionnement Suivi de la cible

Avec les politiques de dimensionnement du suivi des cibles, vous sélectionnez une métrique et définissez une valeur cible. ElastiCache pour Valkey et Redis OSS Auto Scaling crée et gère les CloudWatch alarmes qui déclenchent la politique de dimensionnement et calcule l'ajustement de mise à l'échelle en fonction de la métrique et de la valeur cible. La politique de mise à l'échelle ajoute ou supprime des partitions si nécessaire pour maintenir la métrique à la valeur cible spécifiée ou proche de celle-ci. En plus de maintenir la métrique proche de la valeur cible, une politique de dimensionnement Suivi de la cible s'ajuste également aux fluctuations de la métrique dues à un modèle de charge fluctuant, et minimise les fluctuations rapides dans la capacité du parc.

Par exemple, examinons une politique de mise à l'échelle qui utilise la métrique prédéfinie d'utilisation moyenne de `ElastiCachePrimaryEngineCPUUtilization` avec une valeur cible configurée. Une telle politique peut maintenir l'utilisation du CPU à la valeur cible spécifiée ou proche de celle-ci.

Métriques prédéfinies

Une métrique prédéfinie est une structure qui fait référence à un nom, à une dimension et à une statistique (`average`) spécifiques d'une CloudWatch métrique donnée. Votre politique Auto Scaling définit les métriques prédéfinies suivantes pour votre cluster :

Nom de la métrique prédéfinie	CloudWatch Nom de la métrique	CloudWatch Dimension métrique	Types d'instances non éligibles
`ElastiCachePrimaryEngineCPUUtilization`	`EngineCPUUtilization`	ReplicationGroupId, Rôle = Principal	Aucun
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`	`DatabaseCapacityUsageCountedForEvictPercentage`	Métriques du groupe de réplication Valkey ou Redis OSS	Aucun
`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`	`DatabaseMemoryUsageCountedForEvictPercentage`	Métriques du groupe de réplication Valkey ou Redis OSS	R6gd

Les types d'instance avec hiérarchisation des données ne peuvent pas utiliser `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, car ils stockent des données à la fois dans la mémoire et sur SSD. Le cas de figure attendu pour les instances avec hiérarchisation des données est d'utiliser 100 % de la mémoire et de remplir le SSD si nécessaire.

Critères de Auto Scaling pour les partitions

Lorsque le service détecte que votre métrique prédéfinie est égale ou supérieure au paramètre Target, il augmente automatiquement la capacité de vos partitions. ElastiCache pour Valkey et Redis OSS redimensionne les partitions de votre cluster d'un nombre égal à la plus grande des deux valeurs suivantes : pourcentage de variation par rapport à Target et 20 % des partitions actuelles. Pour la mise à l'échelle, la mise à l'échelle automatique ElastiCache ne sera pas effectuée à moins que la valeur métrique globale ne soit inférieure à 75 % de la cible que vous avez définie.

Pour un exemple de mise à l'échelle, si vous avez 50 partitions et

- si votre Target atteint 30 %, il ElastiCache augmente de 30 %, ce qui se traduit par 65 partitions par cluster.
- si votre Target atteint 10 %, augmente ElastiCache par défaut d'au moins 20 %, ce qui se traduit par 60 partitions par cluster.

Par exemple, si vous avez sélectionné une valeur cible de 60 %, la mise à l'échelle automatique ElastiCache ne sera pas automatique tant que la métrique ne sera pas inférieure ou égale à 45 % (25 % en dessous de la cible de 60 %).

Considérations relatives à Auto Scaling

Gardez les considérations suivantes à l'esprit :

- Une politique de mise à l'échelle Suivi de la cible suppose qu'elle doit effectuer une montée en charge ; lorsque la métrique spécifiée est au-dessus de la valeur cible. Vous ne pouvez pas utiliser une politique de dimensionnement du suivi des cibles pour effectuer une mise à l'échelle lorsque la métrique spécifiée est inférieure à la valeur cible. ElastiCache pour Valkey et Redis OSS redimensionne les partitions d'un écart minimum de 20 % par rapport à la cible des partitions existantes dans le cluster.
- Une politique de suivi des objectifs et d'échelonnement n'effectue pas de mise à l'échelle lorsque la métrique spécifiée a des données insuffisantes. Elle n'effectue pas de mise à l'échelle horizontale, car elle n'interprète pas des données insuffisantes comme une faible utilisation.

- Vous pouvez constater des écarts entre la valeur cible et les points de données de métrique réels. En effet, ElastiCache Auto Scaling agit toujours de manière prudente en arrondissant à la hausse ou à la baisse lorsqu'il détermine la capacité à ajouter ou à supprimer. Cela l'empêche d'ajouter une capacité insuffisante ou de retirer trop de capacité.
- Pour garantir la disponibilité de l'application, le service augmente proportionnellement aux métriques aussi rapidement que possible, mais diminue plus progressivement.
- Vous pouvez avoir plusieurs politiques de dimensionnement du suivi des cibles ElastiCache pour un cluster Valkey et Redis OSS, à condition que chacune d'elles utilise une métrique différente. ElastiCache Auto Scaling a pour objectif de toujours prioriser la disponibilité. Son comportement varie donc selon que les politiques de suivi des cibles sont prêtes à être étendues ou intégrées. Il augmentera la taille du service si l'une des politiques Suivi de la cible est prête pour une augmentation de taille, mais la diminuera uniquement si toutes les politiques Suivi de la cible (avec la portion de diminution en charge activée) sont prêtes pour une diminution de taille.
- Ne modifiez ni ne supprimez les CloudWatch alarmes gérées par ElastiCache Auto Scaling dans le cadre d'une politique de dimensionnement du suivi des cibles. ElastiCache Auto Scaling supprime automatiquement les alarmes lorsque vous supprimez la politique de dimensionnement.
- ElastiCache Auto Scaling ne vous empêche pas de modifier manuellement les partitions de cluster. Ces ajustements manuels n'affectent pas les CloudWatch alarmes existantes associées à la politique de dimensionnement, mais peuvent avoir un impact sur les mesures susceptibles de déclencher ces CloudWatch alarmes.
- Ces CloudWatch alarmes gérées par Auto Scaling sont définies sur la métrique AVG pour toutes les partitions du cluster. Ainsi, avoir des partitions chaudes peut entraîner l'un ou l'autre des scénarios suivants :
  - mise à l'échelle lorsqu'elle n'est pas nécessaire en raison de la charge sur quelques fragments chauds déclenchant une alarme CloudWatch
  - l'absence de mise à l'échelle lorsque cela est nécessaire en raison d'AVG agrégé sur toutes les partitions affectant l'alarme à ne pas briser.
- ElastiCache les limites par défaut relatives aux nœuds par cluster s'appliquent toujours. Ainsi, lorsque vous optez pour Auto Scaling et si vous prévoyez que le nombre maximal de nœuds dépasse la limite par défaut, demandez une augmentation de limite à AWS Service Limits et choisissez le type de limite Nodes per cluster per instance type (Nœuds par cluster par type d'instance).
- Assurez-vous que votre VPC dispose d'un nombre suffisant d'interfaces réseau élastiques ENIs (Elastic Network Interfaces), nécessaires lors du scale-out. Pour de plus amples informations, veuillez consulter Interfaces réseau Elastic.

- Si la capacité disponible n'est pas suffisante EC2, ElastiCache Auto Scaling ne sera pas redimensionné et sera retardé jusqu'à ce que la capacité soit disponible.
- ElastiCache pour Redis OSS, Auto Scaling ne supprimera pas les fragments contenant des emplacements dont la taille d'élément est supérieure à 256 Mo après la sérialisation lors de la scale-in.
- Pendant la réduction de charge, il ne supprimera pas les partitions si la mémoire disponible sur la configuration des partitions résultante est insuffisante.

Ajout d'une politique de mise à l'échelle

Vous pouvez ajouter une politique de dimensionnement à l'aide du AWS Management Console.

Pour ajouter une politique Auto Scaling à un cluster ElastiCache pour Valkey et Redis OSS

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Choisissez le cluster auquel vous voulez ajouter une stratégie (choisissez le nom du cluster et non pas le bouton situé à sa gauche).
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Choisissez add dynamic scaling (ajouter le dimensionnement dynamique).
6. Pour Policy name (Nom de la politique), attribuez un nom de politique.
7. Pour Scalable Dimension (Dimension évolutives) choisir shards (partitions).
8. Pour la métrique cible, choisissez l'une des actions suivantes :
    - Primary CPU Utilization (Utilisation du CPU principal) pour créer une politique basée sur l'utilisation moyenne du CPU.
    - Memory (Mémoire) pour créer une politique basée sur la mémoire de base de données moyenne.
    - Capacité pour créer une politique basée sur l'utilisation moyenne de la capacité de la base de données. La métrique Capacité inclut l'utilisation de la mémoire et du SSD pour les instances avec hiérarchisation des données, ainsi que l'utilisation de la mémoire pour tous les autres types d'instances.

9. Pour la valeur cible, choisissez une valeur supérieure ou égale à 35 et inférieure ou égale à 70. La mise à l'échelle automatique conservera cette valeur pour la métrique cible sélectionnée sur l'ensemble de vos ElastiCache partitions :

   - Utilisation du processeur principal : conserve la valeur cible pour la métrique `EngineCPUUtilization` sur les nœuds primaires.
   - Mémoire : conserve la valeur cible pour la métrique `DatabaseMemoryUsageCountedForEvictPercentage`
   - Capacité conserve la valeur cible pour la métrique `DatabaseCapacityUsageCountedForEvictPercentage`.

   Des partitions de cluster sont ajoutées ou supprimées pour maintenir la métrique proche de la valeur spécifiée.
10. (Facultatif) Les périodes de stabilisation de diminution ou de montée en charge ne sont pas prises en charge à partir de la console. Utilisez le AWS CLI pour modifier les valeurs de recharge.
11. Pour Capacité minimale, saisissez le nombre minimum de partitions que la politique ElastiCache Auto Scaling doit maintenir.
12. Pour Maximum capacity, saisissez le nombre maximum de partitions que la politique ElastiCache Auto Scaling doit gérer. Cette valeur doit être inférieure ou égale à 250.
13. Choisissez Créer.

Enregistrement d'une cible évolutive

Avant de pouvoir utiliser Auto Scaling avec un cluster ElastiCache pour Valkey et Redis OSS, vous devez enregistrer votre cluster avec ElastiCache Auto Scaling. Vous le faites pour définir la dimension de mise à l'échelle et les limites à appliquer à ce cluster. ElastiCache le dimensionnement automatique redimensionne dynamiquement le cluster selon la dimension `elasticache:replication-group:NodeGroups` évolutive, qui représente le nombre de partitions du cluster.

En utilisant le AWS CLI

Pour enregistrer votre cluster ElastiCache for Valkey et Redis OSS, utilisez la register-scalable-targetcommande avec les paramètres suivants :

- `--service-namespace` – Définissez cette valeur sur `elasticache`

- `--resource-id`— L'identifiant de ressource du cluster. Pour ce paramètre, le type de ressource est `ReplicationGroup` et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- `--scalable-dimension` – Définissez cette valeur sur `elasticache:replication-group:NodeGroups`.
- `--max-capacity` — Le nombre maximum de partitions à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre `--min-capacity`, `--max-capacity` et le nombre de partitions dans votre cluster, veuillez consulter Capacité minimale et maximale.
- `--min-capacity` — Le nombre minimum de partitions à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre `--min-capacity`, `--max-capacity` et le nombre de partitions dans votre cluster, veuillez consulter Capacité minimale et maximale.

Example

Dans l'exemple suivant, vous enregistrez un ElastiCache cluster nommé`myscalablecluster`. L'enregistrement indique que le cluster doit être mis à l'échelle de façon dynamique pour contenir de une à huit partitions.

Pour Linux, macOS ou Unix :

```
aws application-autoscaling register-scalable-target \
    --service-namespace elasticache \
    --resource-id replication-group/myscalablecluster \
    --scalable-dimension elasticache:replication-group:NodeGroups \
    --min-capacity 1 \
    --max-capacity 10 \
```

Pour Windows :

```
aws application-autoscaling register-scalable-target ^
    --service-namespace elasticache ^
    --resource-id replication-group/myscalablecluster ^
    --scalable-dimension elasticache:replication-group:NodeGroups ^
    --min-capacity 1 ^
    --max-capacity 10 ^
```

Utilisation de l'API

Pour enregistrer votre ElastiCache cluster, utilisez la register-scalable-targetcommande avec les paramètres suivants :

- ServiceNamespace — Définissez cette valeur sur elasticache.
- ResourceID — Identifiant de ressource pour le cluster. ElastiCache Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- ScalableDimension — Définissez cette valeur sur`elasticache:replication-group:NodeGroups`.
- MinCapacity — Le nombre minimum de partitions à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.
- MaxCapacity — Le nombre maximum de partitions à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.

Example

Dans l'exemple suivant, vous enregistrez un ElastiCache cluster nommé `myscalablecluster` avec l'API Application Auto Scaling. Cet enregistrement indique que le cluster doit être dimensionné de façon dynamique pour contenir de un à 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups",
    "MinCapacity": 1,
    "MaxCapacity": 5
```

```
}
```

Définition d'une politique de mise à l'échelle

Une configuration de politique de dimensionnement Suivi de la cible est représentée par un bloc JSON dans lequel sont définies les métriques et valeurs cibles. Vous pouvez enregistrer une configuration de politique de dimensionnement sous forme de bloc JSON dans un fichier texte. Vous utilisez ce fichier texte lorsque vous appelez l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling. Pour plus d'informations sur la syntaxe de la configuration d'une politique, veuillez consulter TargetTrackingScalingPolicyConfiguration dans la Référence de l'API Application Auto Scaling.

Les options suivantes sont disponibles pour définir une configuration de politique de suivi de cible et d'échelonnement :

Rubriques

- Utilisation d'une métrique prédéfinie
- Utilisation d'une métrique personnalisée
- Utilisation des temps de stabilisation

Utilisation d'une métrique prédéfinie

En utilisant des métriques prédéfinies, vous pouvez définir rapidement une politique de dimensionnement pour le suivi des cibles ElastiCache pour un cluster Valkey et Redis OSS qui fonctionne avec le suivi des cibles dans Auto Scaling. ElastiCache

Actuellement, ElastiCache prend en charge les métriques prédéfinies suivantes dans NodeGroup Auto Scaling :

- ElastiCachePrimaryEngineCPUUtilization— La valeur moyenne de la `EngineCPUUtilization` métrique pour CloudWatch tous les nœuds principaux du cluster.
- ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage— La valeur moyenne de la `DatabaseMemoryUsageCountedForEvictPercentage` métrique pour CloudWatch tous les nœuds principaux du cluster.
- ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage— La valeur moyenne de la `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` métrique pour CloudWatch tous les nœuds principaux du cluster.

Pour plus d'informations sur les métriques `EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` et `DatabaseCapacityUsageCountedForEvictPercentage`, consultez Surveillance de l'utilisation à l'aide de CloudWatch métriques. Pour utiliser une métrique prédéfinie dans votre politique de dimensionnement, créez une configuration de suivi de la cible pour votre politique de dimensionnement. Cette configuration doit inclure `PredefinedMetricSpecification` pour la métrique prédéfinie et TargetValue pour la valeur cible de cette métrique.

Example

L'exemple suivant décrit une configuration de politique typique pour le dimensionnement du suivi des cibles ElastiCache pour un cluster Valkey et Redis OSS. Dans cette configuration, la métrique `ElastiCachePrimaryEngineCPUUtilization` prédéfinie est utilisée pour ajuster le cluster en fonction d'une utilisation moyenne du processeur de 40 % sur tous les nœuds principaux du cluster.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    }
}
```

Utilisation d'une métrique personnalisée

L'utilisation de métriques personnalisées vous permet de définir une politique de dimensionnement Suivi de la cible répondant à vos exigences personnelles. Vous pouvez définir une métrique personnalisée en fonction d'une métrique ElastiCache qui évolue proportionnellement à la mise à l'échelle. Toutes les ElastiCache mesures ne fonctionnent pas pour le suivi des cibles. La métrique doit être une métrique d'utilisation valide et décrire le degré d'occupation d'une instance. La valeur de la métrique doit augmenter ou diminuer proportionnellement au nombre de partitions dans le cluster. Cette augmentation ou diminution proportionnelle est nécessaire pour que les données de la métrique puissent être utilisées afin d'augmenter ou de réduire proportionnellement le nombre de partitions.

Example

L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, une métrique personnalisée ajuste un cluster

ElastiCache pour Redis OSS en fonction d'une utilisation moyenne du processeur de 50 % sur toutes les partitions d'un cluster nommé. `my-db-cluster`

```
{
    "TargetValue": 50,
    "CustomizedMetricSpecification":
    {
        "MetricName": "EngineCPUUtilization",
        "Namespace": "AWS/ElastiCache",
        "Dimensions": [
            {
                "Name": "ReplicationGroup","Value": "my-db-cluster"
            },
            {
                "Name": "Role","Value": "PRIMARY"
            }
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Utilisation des temps de stabilisation

Vous pouvez spécifier une valeur, en secondes, pour que `ScaleOutCooldown` ajoute un temps de stabilisation à la montée en puissance de votre cluster. De la même manière, vous pouvez ajouter une valeur, en secondes, pour que `ScaleInCooldown` ajoute un temps de stabilisation pour la diminution de charge de votre cluster. Pour plus d'informations, reportez-vous TargetTrackingScalingPolicyConfigurationà la section Application Auto Scaling API Reference.

L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, la métrique `ElastiCachePrimaryEngineCPUUtilization` prédéfinie est utilisée ElastiCache pour ajuster un cluster Redis OSS en fonction d'une utilisation moyenne du processeur de 40 % sur tous les nœuds principaux de ce cluster. La configuration indique un temps de stabilisation de diminution en charge de 10 minutes et un temps de stabilisation de montée en charge de 5 minutes.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
```

```
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Désactivation de l'activité de diminution en charge

Vous pouvez empêcher la configuration de la politique de dimensionnement du suivi des cibles de s'étendre dans votre cluster en désactivant l'activité de dimensionnement. La désactivation de l'activité de diminution en charge empêche la politique de mise à l'échelle de supprimer des partitions, tout en l'autorisant encore à les créer si nécessaire.

Vous pouvez spécifier une valeur booléenne pour que `DisableScaleIn` active ou désactive l'activité de diminution en charge de votre cluster. Pour plus d'informations, reportez-vous TargetTrackingScalingPolicyConfigurationà la section Application Auto Scaling API Reference.

L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, la métrique `ElastiCachePrimaryEngineCPUUtilization` prédéfinie ajuste un cluster ElastiCache pour Valkey et Redis OSS en fonction d'une utilisation moyenne du processeur de 40 % sur tous les nœuds principaux de ce cluster. La configuration désactive l'activité de diminution en charge pour la politique de dimensionnement.

```
{
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
        "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Application d'une politique de mise à l'échelle

Après avoir enregistré votre cluster auprès ElastiCache de Valkey et Redis OSS Auto Scaling et défini une politique de dimensionnement, vous appliquez la politique de dimensionnement au cluster enregistré. Pour appliquer une politique de dimensionnement à un ElastiCache cluster Redis OSS, vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling.

Appliquer une politique de dimensionnement à l'aide du AWS CLI

Pour appliquer une politique de dimensionnement à votre cluster ElastiCache for Valkey et Redis OSS, utilisez la put-scaling-policycommande avec les paramètres suivants :

- --policy-name – Nom de la politique de mise à l'échelle.
- --policy-type – Définissez cette valeur à `TargetTrackingScaling`.
- --resource-id — L'identifiant de la ressource. Pour ce paramètre, le type de ressource est `ReplicationGroup` et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace – Définissez cette valeur à `elasticache`.
- --scalable-dimension – Définissez cette valeur à `elasticache:replication-group:NodeGroups`.
- -- target-tracking-scaling-policy -configuration — Configuration de la politique de dimensionnement du suivi des cibles à utiliser pour le cluster.

Dans l'exemple suivant, vous appliquez une politique de dimensionnement de suivi des cibles nommée `myscalablepolicy` à un cluster ElastiCache pour Valkey et Redis OSS nommé auto scaling. `myscalablecluster` ElastiCache Pour ce faire, vous utilisez une configuration de politique enregistrée dans un fichier nommé `config.json`.

Pour Linux, macOS ou Unix :

```
aws application-autoscaling put-scaling-policy \
    --policy-name myscalablepolicy \
    --policy-type TargetTrackingScaling \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:NodeGroups \
    --target-tracking-scaling-policy-configuration file://config.json
```

Pour Windows :

```
aws application-autoscaling put-scaling-policy ^
    --policy-name myscalablepolicy ^
    --policy-type TargetTrackingScaling ^
```

```
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:NodeGroups ^
    --target-tracking-scaling-policy-configuration file://config.json
```

Application d'une politique de mise à l'échelle à l'aide de l'API

Pour appliquer une politique de dimensionnement à votre cluster ElastiCache for Valkey et Redis OSS, utilisez la PutScalingPolicy AWS CLI commande avec les paramètres suivants :

- --policy-name – Nom de la politique de mise à l'échelle.
- --resource-id — L'identifiant de la ressource. Pour ce paramètre, le type de ressource est `ReplicationGroup` et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace – Définissez cette valeur à `elasticache`.
- --scalable-dimension – Définissez cette valeur à `elasticache:replication-group:NodeGroups`.
- -- target-tracking-scaling-policy -configuration — Configuration de la politique de dimensionnement du suivi des cibles à utiliser pour le cluster.

Dans l'exemple suivant, vous appliquez une politique de dimensionnement de suivi des cibles nommée `myscalablepolicy` à un ElastiCache cluster nommé avec mise à l'échelle `myscalablecluster` ElastiCache automatique. Vous utilisez une configuration de politique basée sur la métrique prédéfinie `ElastiCachePrimaryEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups",
```

```
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration": {
        "TargetValue": 40.0,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
        }
    }
}
```

Modification d'une politique de dimensionnement

Vous pouvez modifier une politique de dimensionnement à l'aide de l' AWS Management Console API Application Auto Scaling ou de l'API Application Auto Scaling. AWS CLI

Modification d'une politique de dimensionnement à l'aide du AWS Management Console

Pour modifier une politique Auto Scaling pour un cluster ElastiCache pour Valkey et Redis OSS

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez le moteur approprié.
3. Choisissez le cluster auquel vous voulez ajouter une stratégie (choisissez le nom du cluster et non pas le bouton situé à sa gauche).
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Sous Scaling policies (Stratégies de dimensionnement), choisissez le bouton à gauche de la stratégie Auto Scaling que vous voulez changer, puis choisissez Modify (Modifier).
6. Apportez les modifications nécessaires à la politique.
7. Sélectionnez Modifier.

Modification d'une politique de dimensionnement à l'aide de l'API AWS CLI and

Vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling pour modifier une politique de dimensionnement de la même manière que vous appliquez une politique de dimensionnement :

- Lorsque vous utilisez le AWS CLI, spécifiez le nom de la politique que vous souhaitez modifier dans le `--policy-name` paramètre. Spécifiez de nouvelles valeurs pour les paramètres que vous souhaitez modifier.

- Lorsque vous utilisez l'API Application Auto Scaling, spécifiez le nom de la politique à modifier dans le paramètre `PolicyName`. Spécifiez de nouvelles valeurs pour les paramètres que vous souhaitez modifier.

Pour de plus amples informations, veuillez consulter Application d'une politique de mise à l'échelle.

Suppression d'une politique de dimensionnement

Vous pouvez supprimer une politique de dimensionnement à l'aide de l' AWS Management Console API Application Auto Scaling ou de l'API Application Auto Scaling. AWS CLI

Suppression d'une politique de dimensionnement à l'aide du AWS Management Console

Pour supprimer une politique Auto Scaling ElastiCache pour un cluster Redis OSS

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Choisissez le cluster dont vous voulez modifier la stratégie Auto Scaling (choisissez le nom du cluster et non pas le bouton situé à sa gauche).
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Sous Scaling policies (Stratégies de dimensionnement), choisissez la stratégie Auto Scaling, puis choisissez Delete (Supprimer).

Suppression d'une politique de dimensionnement à l'aide du AWS CLI

Pour supprimer une politique de dimensionnement de votre cluster ElastiCache pour Valkey et Redis OSS, utilisez la delete-scaling-policy AWS CLI commande avec les paramètres suivants :

- --policy-name – Nom de la politique de mise à l'échelle.
- --resource-id — L'identifiant de la ressource. Pour ce paramètre, le type de ressource est `ReplicationGroup` et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace – Définissez cette valeur à `elasticache`.
- --scalable-dimension – Définissez cette valeur à `elasticache:replication-group:NodeGroups`.

Dans l'exemple suivant, vous supprimez une politique de dimensionnement de suivi des cibles nommée à `myscalablepolicy` partir d'un cluster nommé. `myscalablecluster`

Pour Linux, macOS ou Unix :

```
aws application-autoscaling delete-scaling-policy \
    --policy-name myscalablepolicy \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:NodeGroups
```

Pour Windows :

```
aws application-autoscaling delete-scaling-policy ^
    --policy-name myscalablepolicy ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:NodeGroups
```

Suppression d'une politique de mise à l'échelle à l'aide de l'API

Pour supprimer une politique de dimensionnement de votre cluster ElastiCache pour Valkey et Redis OSS, utilisez la DeleteScalingPolicy AWS CLI commande avec les paramètres suivants :

- --policy-name – Nom de la politique de mise à l'échelle.
- --resource-id — L'identifiant de la ressource. Pour ce paramètre, le type de ressource est `ReplicationGroup` et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace – Définissez cette valeur à `elasticache`.
- --scalable-dimension – Définissez cette valeur à `elasticache:replication-group:NodeGroups`.

Dans l'exemple suivant, vous supprimez une politique de dimensionnement de suivi des cibles nommée à `myscalablepolicy` partir d'un cluster nommé. `myscalablecluster`

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
```

```
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:NodeGroups"
}
```

Utilisation AWS CloudFormation pour les politiques Auto Scaling

Cet extrait montre comment créer une politique de suivi cible et l'appliquer à une AWS::ElastiCache::ReplicationGroupressource utilisant cette ressource. AWS::ApplicationAutoScaling::ScalableTarget Elle utilise les fonctions intrinsèques Fn::Join et Ref pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::ElastiCache::ReplicationGroup` qui est spécifiée dans le même modèle.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 3
     MinCapacity: 1
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:NodeGroups'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

  ScalingPolicy:
    Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
    Properties:
      ScalingTargetId: !Ref ScalingTarget
      ServiceNamespace: elasticache
      PolicyName: testpolicy
      PolicyType: TargetTrackingScaling
      ScalableDimension: 'elasticache:replication-group:NodeGroups'
      TargetTrackingScalingPolicyConfiguration:
        PredefinedMetricSpecification:
          PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
```

```
        TargetValue: 40
```

Mise à l'échelle planifiée

La mise à l'échelle en fonction d'une planification vous permet de mettre à l'échelle l'application en réponse aux changements de demande. Pour utiliser le dimensionnement planifié, vous créez des actions planifiées qui indiquent ElastiCache à Valkey et Redis OSS d'effectuer des activités de dimensionnement à des moments précis. Lorsque vous créez une action planifiée, vous spécifiez un cluster existant, le moment où l'activité de dimensionnement doit avoir lieu, la capacité minimale et la capacité maximale. Vous pouvez créer des actions planifiées pour une mise à l'échelle unique ou selon une planification récurrente.

Vous ne pouvez créer une action planifiée que pour les clusters déjà existants. Vous ne pouvez pas créer une action planifiée en même temps que vous créez un cluster.

Pour plus d'informations sur la terminologie relative à la création, à la gestion et à la suppression d'actions planifiées, veuillez consulter Commandes généralement utilisées pour la création, la gestion et la suppression d'actions planifiées

Pour créer selon un calendrier récurrent :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Choisissez le cluster auquel vous souhaitez ajouter une politique.
4. Cliquez sur l'onglet Manage Auto Scaling policies (Gestion des politiques Auto Scaling) dans la liste déroulante des Actions.
5. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
6. Dans Auto Scaling policies (Politiques de scalabilité automatique), la boîte de dialogue Add Scaling policy (Ajouter une politique de mise à l'échelle) s'affiche. Choisissez Scheduled scaling (Mise à l'échelle planifiée).
7. Pour Policy name (Nom de la politique), saisissez un nom de politique.
8. Pour Scalable Dimension (Dimension évolutives) choisir shards (partitions).
9. Pour Target Shards (Partitions cibles), choisissez la valeur.
10. Pour Recurrence (Récurrence), choisissezRecurring (Récurrents).
11. Pour Frequency (Fréquence), choisissez la valeur respective.

12. Pour Start Date (Date de début) et Start Time (Heure de début), choisissez l'heure à partir de laquelle la politique entrera en vigueur.
13. Choisissez Add policy (Ajouter la politique).

Pour créer une action planifiée unique :

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Choisissez le cluster auquel vous souhaitez ajouter une politique.
4. Cliquez sur l'onglet Manage Auto Scaling policies (Gestion des politiques Auto Scaling) dans la liste déroulante des Actions.
5. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
6. Dans Auto Scaling policies (Politiques de scalabilité automatique), la boîte de dialogue Add Scaling policy (Ajouter une politique de mise à l'échelle) s'affiche. Choisissez Scheduled scaling (Mise à l'échelle planifiée).
7. Pour Policy name (Nom de la politique), saisissez un nom de politique.
8. Pour Scalable Dimension (Dimension évolutives) choisir shards (partitions).
9. Pour Target Shards (Partitions cibles), choisissez la valeur.
10. Pour Recurrence (Récurrence), choisissez One Time (Une fois).
11. Pour Start Date (Date de début) et Start Time (Heure de début), choisissez l'heure à partir de laquelle la politique entrera en vigueur.
12. Pour End Date (Date de fin) choisir la date jusqu'à laquelle la police sera en vigueur.
13. Choisissez Add policy (Ajouter la politique).

Pour supprimer une action planifiée

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Choisissez le cluster auquel vous souhaitez ajouter une politique.
4. Cliquez sur l'onglet Manage Auto Scaling policies (Gestion des politiques Auto Scaling) dans la liste déroulante des Actions.

5. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
6. Dans la section Auto scaling policies (Politiques Auto Scaling), choisissez la politique Auto Scaling, puis Delete (Supprimer) depuis la fenêtre Actions.

Pour gérer la mise à l'échelle planifiée à l'aide de la AWS CLI

Utilisez la mise à l'échelle automatique des applications APIs suivante :

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

Utiliser AWS CloudFormation pour créer une action planifiée

Cet extrait montre comment créer une politique de suivi cible et l'appliquer à une AWS::ElastiCache::ReplicationGroupressource utilisant cette ressource. AWS::ApplicationAutoScaling::ScalableTarget Elle utilise les fonctions intrinsèques Fn::Join et Ref pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::ElastiCache::ReplicationGroup` qui est spécifiée dans le même modèle.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 3
     MinCapacity: 1
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:NodeGroups'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
     ScheduledActions:
       - EndTime: '2020-12-31T12:00:00.000Z'
         ScalableTargetAction:
           MaxCapacity: '5'
           MinCapacity: '2'
         ScheduledActionName: First
         Schedule: 'cron(0 18 * * ? *)'
```

Utilisation d'Auto Scaling avec des réplicas

Un groupe de ElastiCache réplication peut configurer un ou plusieurs caches pour qu'ils fonctionnent comme un seul nœud logique.

Vous trouverez ci-dessous des informations sur le suivi des cibles et les politiques planifiées, ainsi que sur la manière de les appliquer à l'aide du AWS Management Console AWS CLI et APIs.

Politiques de dimensionnement Suivi de la cible

Avec les politiques de dimensionnement du suivi des cibles, vous sélectionnez une métrique et définissez une valeur cible. ElastiCache pour Valkey et Redis OSS AutoScaling crée et gère les CloudWatch alarmes qui déclenchent la politique de dimensionnement et calcule l'ajustement de mise à l'échelle en fonction de la métrique et de la valeur cible. La politique de mise à l'échelle ajoute ou supprime des partitions si nécessaire pour maintenir la métrique à la valeur cible spécifiée ou proche de celle-ci. En plus de maintenir la métrique proche de la valeur cible, une politique de dimensionnement Suivi de la cible s'ajuste également aux fluctuations de la métrique dues à un modèle de charge fluctuant, et minimise les fluctuations rapides dans la capacité du parc.

Critères Auto Scaling pour les réplicas

Votre politique Auto Scaling définit la métrique prédéfinie suivante pour votre cluster :

`ElastiCacheReplicaEngineCPUUtilization`: le seuil d'utilisation du processeur AVG EngineCPU agrégé pour toutes les répliques utilisées ElastiCache pour déclencher une opération d'auto-scaling. Vous pouvez définir la cible d'utilisation entre 35 et 70 %.

Lorsque le service détecte que votre `ElastiCacheReplicaEngineCPUUtilization` métrique est égale ou supérieure au paramètre Target, il augmente automatiquement le nombre de répliques sur vos partitions. ElastiCache redimensionne les répliques de votre cluster d'un nombre égal à la plus grande des deux valeurs suivantes : variation en pourcentage par rapport à Target et une réplique. Pour la mise à l'échelle, la mise à l'échelle automatique ElastiCache ne sera pas effectuée à moins que la valeur métrique globale ne soit inférieure à 75 % de la cible que vous avez définie.

Pour un exemple de montée en puissance, si vous avez 5 partitions et 1 réplica chacun :

Si votre Target atteint 30 %, ElastiCache pour Valkey et Redis, OSS augmente d'une réplique (maximum (0,3, par défaut 1)) sur toutes les partitions, ce qui donne 5 partitions avec 2 répliques chacune,

Par exemple, si vous avez sélectionné une valeur cible de 60 %, ElastiCache pour Valkey et Redis, OSS ne procédera pas à une mise à l'échelle automatique tant que la métrique ne sera pas inférieure ou égale à 45 % (25 % en dessous de la cible de 60 %).

Considérations relatives à Auto Scaling

Gardez les considérations suivantes à l'esprit :

- Une politique de mise à l'échelle Suivi de la cible suppose qu'elle doit effectuer une montée en charge ; lorsque la métrique spécifiée est au-dessus de la valeur cible. Vous ne pouvez pas utiliser une politique de dimensionnement du suivi des cibles pour effectuer une mise à l'échelle lorsque la métrique spécifiée est inférieure à la valeur cible. ElastiCache pour Valkey et Redis OSS redimensionne les répliques au maximum (% d'écart arrondi par rapport à Target, 1 par défaut) des répliques existantes sur toutes les partitions du cluster.
- Une politique de suivi des objectifs et d'échelonnement n'effectue pas de mise à l'échelle lorsque la métrique spécifiée a des données insuffisantes. Elle n'effectue pas de mise à l'échelle horizontale car elle n'interprète pas des données insuffisantes comme une faible utilisation.
- Vous pouvez constater des écarts entre la valeur cible et les points de données de métrique réels. En effet, ElastiCache Auto Scaling agit toujours de manière prudente en arrondissant à la hausse ou à la baisse lorsqu'il détermine la capacité à ajouter ou à supprimer. Cela l'empêche d'ajouter une capacité insuffisante ou de retirer trop de capacité.
- Pour garantir la disponibilité de l'application, le service augmente proportionnellement aux métriques aussi rapidement que possible, mais diminue plus progressivement avec une augmentation maximale d'un réplica sur toutes les partitions du cluster.
- Vous pouvez avoir plusieurs politiques de dimensionnement du suivi des cibles ElastiCache pour un cluster Valkey et Redis OSS, à condition que chacune d'elles utilise une métrique différente. Auto Scaling a pour objectif de toujours prioriser la disponibilité. Son comportement varie donc selon que les politiques de suivi des cibles sont prêtes à être étendues ou intégrées. Il augmentera la taille du service si l'une des politiques Suivi de la cible est prête pour une augmentation de taille, mais la diminuera uniquement si toutes les politiques Suivi de la cible (avec la portion de diminution en charge activée) sont prêtes pour une diminution de taille.
- Ne modifiez ni ne supprimez les CloudWatch alarmes gérées par ElastiCache Auto Scaling dans le cadre d'une politique de dimensionnement du suivi des cibles. Auto Scaling supprime automatiquement les alarmes lorsque vous supprimez la politique de dimensionnement ou lorsque vous supprimez le cluster.
- ElastiCache Auto Scaling ne vous empêche pas de modifier manuellement les répliques entre les partitions. Ces ajustements manuels n'affectent pas les CloudWatch alarmes existantes associées

à la politique de dimensionnement, mais peuvent avoir un impact sur les mesures susceptibles de déclencher ces CloudWatch alarmes.

- Ces CloudWatch alarmes gérées par Auto Scaling sont définies sur la métrique AVG pour toutes les partitions du cluster. Ainsi, avoir des partitions chaudes peut entraîner l'un ou l'autre des scénarios suivants :
  - mise à l'échelle lorsqu'elle n'est pas nécessaire en raison de la charge sur quelques fragments chauds déclenchant une alarme CloudWatch
  - l'absence de mise à l'échelle lorsque cela est nécessaire en raison d'AVG agrégé sur toutes les partitions affectant l'alarme à ne pas briser.
- ElastiCache les limites par défaut relatives aux nœuds par cluster s'appliquent toujours. Ainsi, lorsque vous optez pour Auto Scaling et si vous prévoyez que le nombre maximal de nœuds dépasse la limite par défaut, demandez une augmentation de limite à AWS Service Limits et choisissez le type de limite Nodes per cluster per instance type (Nœuds par cluster par type d'instance).
- Assurez-vous que votre VPC dispose d'un nombre suffisant d'interfaces réseau élastiques ENIs (Elastic Network Interfaces), nécessaires lors du scale-out. Pour de plus amples informations, veuillez consulter Interfaces réseau Elastic.
- Si la capacité disponible n'est pas suffisante EC2, ElastiCache Auto Scaling n'augmentera pas tant que la capacité ne sera pas disponible ou si vous modifiez manuellement le cluster en fonction des types d'instances dotés d'une capacité suffisante.
- ElastiCache Auto Scaling ne prend pas en charge le dimensionnement des répliques avec un cluster contenant `ReservedMemoryPercent` moins de 25 %. Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Ajout d'une politique de mise à l'échelle

Vous pouvez ajouter une politique de dimensionnement à l'aide du AWS Management Console.

Ajout d'une politique de dimensionnement à l'aide du AWS Management Console

Pour ajouter une politique de dimensionnement automatique à ElastiCache Valkey et Redis OSS

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.

3. Choisissez le cluster auquel vous voulez ajouter une stratégie (choisissez le nom du cluster et non pas le bouton situé à sa gauche).
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Choisissez add dynamic scaling (ajouter le dimensionnement dynamique).
6. Sous Scaling policies (Stratégies de dimensionnement), choisissez Add dynamic scaling (Ajouter le dimensionnement dynamique).
7. Pour Policy name (Nom de la politique), saisissez un nom de politique.
8. Pour Scalable Dimension (Dimension évolutives), sélectionnez Replicas (Réplicas) dans la boîte de dialogue.
9. Pour la valeur cible, saisissez le pourcentage moyen d'utilisation du processeur que vous souhaitez conserver sur les ElastiCache répliques. Cette valeur doit être >=35 et <=70. Des réplicas de cluster sont ajoutés ou supprimés pour maintenir la métrique proche de la valeur spécifiée.
10. (Facultatif) les périodes de stabilisation de réduction ou de montée en charge ne sont pas prises en charge par la console. Utilisez le AWS CLI pour modifier les valeurs de refroidissement.
11. Pour Capacité minimale, saisissez le nombre minimum de répliques que la politique ElastiCache Auto Scaling doit maintenir.
12. Dans Maximum capacity, saisissez le nombre maximum de répliques que la politique ElastiCache Auto Scaling doit gérer. La valeur doit être >=5.
13. Choisissez Créer.

Enregistrement d'une cible évolutive

Vous pouvez appliquer une politique de dimensionnement basée sur une métrique prédéfinie ou personnalisée. Pour ce faire, vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling. La première étape consiste à enregistrer votre groupe de réplication ElastiCache pour Valkey et Redis OSS auprès d'Auto Scaling.

Avant de pouvoir utiliser la mise à l'échelle ElastiCache automatique avec un cluster, vous devez enregistrer votre cluster auprès du dimensionnement ElastiCache automatique. Vous le faites pour définir la dimension de mise à l'échelle et les limites à appliquer à ce cluster. ElastiCache le dimensionnement automatique redimensionne dynamiquement le cluster selon la dimension `elasticache:replication-group:Replicas` évolutive, qui représente le nombre de répliques de cluster par partition.

Utilisation de la CLI

Pour enregistrer votre ElastiCache cluster, utilisez la register-scalable-targetcommande avec les paramètres suivants :

- --service-namespace : définissez cette valeur à elasticache.
- --resource-id — Identifiant de ressource pour le cluster. ElastiCache Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --scalable-dimension : définit cette valeur à `elasticache:replication-group:Replicas`.
- --min-capacity — Le nombre minimum de répliques à gérer par le dimensionnement automatique. ElastiCache Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.
- --max-capacity — Le nombre maximum de répliques à gérer par le dimensionnement automatique. ElastiCache Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.

Example

Dans l'exemple suivant, vous enregistrez un ElastiCache cluster nommé`myscalablecluster`. L'enregistrement indique que le cluster doit être dimensionné de façon dynamique pour contenir de un à 5 réplicas.

Pour Linux, macOS ou Unix :

```
aws application-autoscaling register-scalable-target \
    --service-namespace elasticache \
    --resource-id replication-group/myscalablecluster \
    --scalable-dimension elasticache:replication-group:Replicas \
    --min-capacity 1 \
    --max-capacity 5 \
```

Pour Windows :

```
aws application-autoscaling register-scalable-target ^
    --service-namespace elasticache ^
    --resource-id replication-group/myscalablecluster ^
    --scalable-dimension elasticache:replication-group:Replicas ^
    --min-capacity 1 ^
    --max-capacity 5 ^
```

Utilisation de l'API

Pour enregistrer votre ElastiCache cluster, utilisez la register-scalable-targetcommande avec les paramètres suivants :

- ServiceNamespace — Définissez cette valeur sur elasticache.
- ResourceID — Identifiant de ressource pour le cluster. ElastiCache Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- ScalableDimension — Définissez cette valeur sur`elasticache:replication-group:Replicas`.
- MinCapacity — Le nombre minimum de répliques à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.
- MaxCapacity — Le nombre maximum de répliques à gérer par le dimensionnement ElastiCache automatique. Pour plus d'informations sur la relation entre --min-capacity, --max-capacity et le nombre de réplicas dans votre cluster, veuillez consulter Capacité minimale et maximale.

Example

Dans l'exemple suivant, vous enregistrez un cluster nommé `myscalablecluster` avec l'API Application Auto Scaling. Cet enregistrement indique que le cluster doit être dimensionné de façon dynamique pour contenir de un à 5 réplicas.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas",
    "MinCapacity": 1,
    "MaxCapacity": 5
```

```
}
```

Définition d'une politique de mise à l'échelle

Une configuration de politique de dimensionnement Suivi de la cible est représentée par un bloc JSON dans lequel sont définies les métriques et valeurs cibles. Vous pouvez enregistrer une configuration de politique de dimensionnement sous forme de bloc JSON dans un fichier texte. Vous utilisez ce fichier texte lorsque vous appelez l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling. Pour plus d'informations sur la syntaxe de la configuration d'une stratégie, consultez TargetTrackingScalingPolicyConfiguration dans le manuel Référence d'API Application Auto Scaling.

Les options suivantes sont disponibles pour définir une configuration de politique de suivi de cible et d'échelonnement :

Rubriques

- Utilisation d'une métrique prédéfinie
- Modification d'une politique de dimensionnement
- Suppression d'une politique de dimensionnement
- Utilisation AWS CloudFormation pour les politiques Auto Scaling
- Mise à l'échelle planifiée

Utilisation d'une métrique prédéfinie

Une configuration de politique de dimensionnement Suivi de la cible est représentée par un bloc JSON dans lequel sont définies les métriques et valeurs cibles. Vous pouvez enregistrer une configuration de politique de dimensionnement sous forme de bloc JSON dans un fichier texte. Vous utilisez ce fichier texte lorsque vous appelez l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling. Pour plus d'informations sur la syntaxe de la configuration d'une stratégie, consultez TargetTrackingScalingPolicyConfiguration dans le manuel Référence d'API Application Auto Scaling.

Les options suivantes sont disponibles pour définir une configuration de politique de suivi de cible et d'échelonnement :

Rubriques

- Utilisation d'une métrique prédéfinie

- Utilisation d'une métrique personnalisée
- Utilisation des temps de stabilisation
- Désactivation de l'activité de diminution en charge
- Appliquer une politique de dimensionnement à un cluster ElastiCache pour Valkey et Redis OSS

Utilisation d'une métrique prédéfinie

En utilisant des métriques prédéfinies, vous pouvez définir rapidement une politique de dimensionnement pour le suivi des cibles ElastiCache pour un cluster Valkey et Redis OSS qui fonctionne avec le suivi des cibles dans Auto Scaling. ElastiCache Actuellement, ElastiCache prend en charge la métrique prédéfinie suivante dans ElastiCache Replicas Auto Scaling :

`ElastiCacheReplicaEngineCPUUtilization`— La valeur moyenne de la CPUUtilization métrique Engine pour CloudWatch toutes les répliques du cluster. Vous trouverez la valeur de la métrique agrégée ci-dessous ElastiCache `ReplicationGroupId, Role` pour CloudWatch Required ReplicationGroupId et Role Replica.

Pour utiliser une métrique prédéfinie dans votre politique de dimensionnement, créez une configuration de suivi de la cible pour votre politique de dimensionnement. Cette configuration doit inclure `PredefinedMetricSpecification` pour la métrique prédéfinie et `TargetValue` pour la valeur cible de cette métrique.

Utilisation d'une métrique personnalisée

L'utilisation de métriques personnalisées vous permet de définir une stratégie de dimensionnement Suivi de la cible répondant à vos exigences personnelles. Vous pouvez définir une métrique personnalisée basée sur n'importe quelle métrique ElastiCache pour Valkey et Redis OSS qui change proportionnellement à la mise à l'échelle. Toutes les ElastiCache mesures ne fonctionnent pas pour le suivi des cibles. La métrique doit être une métrique d'utilisation valide et décrire le degré d'occupation d'une instance. La valeur de la métrique doit augmenter ou diminuer proportionnellement au nombre de réplicas dans le cluster. Cette augmentation ou diminution proportionnelle est nécessaire pour que les données de la métrique puissent être utilisées afin d'augmenter ou de réduire proportionnellement le nombre de réplicas .

Example

L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, une métrique personnalisée ajuste un cluster en fonction

d'une utilisation moyenne du processeur de 50 % pour toutes les répliques d'un cluster nommé. `my-db-cluster`

```
{"TargetValue": 50,
    "CustomizedMetricSpecification":
    {"MetricName": "EngineCPUUtilization",
        "Namespace": "AWS/ElastiCache",
        "Dimensions": [
            {"Name": "ReplicationGroup","Value": "my-db-cluster"},
            {"Name": "Role","Value": "REPLICA"}
        ],
        "Statistic": "Average",
        "Unit": "Percent"
    }
}
```

Utilisation des temps de stabilisation

Vous pouvez spécifier une valeur, en secondes, pour que `ScaleOutCooldown` ajoute un temps de stabilisation à la montée en puissance de votre cluster. De la même manière, vous pouvez ajouter une valeur, en secondes, pour que `ScaleInCooldown` ajoute un temps de stabilisation pour la diminution de charge de votre cluster. Pour plus d'informations sur `ScaleInCooldown` et `ScaleOutCooldown`, consultez TargetTrackingScalingPolicyConfiguration dans le manuel Référence d'API Application Auto Scaling. L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, la métrique `ElastiCacheReplicaEngineCPUUtilization` prédéfinie est utilisée pour ajuster un cluster en fonction d'une utilisation moyenne du processeur de 40 % pour toutes les répliques de ce cluster. La configuration indique un temps de stabilisation de diminution en charge de 10 minutes et un temps de stabilisation de montée en charge de 5 minutes.

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "ScaleInCooldown": 600,
    "ScaleOutCooldown": 300
}
```

Désactivation de l'activité de diminution en charge

Vous pouvez empêcher la configuration de la politique de dimensionnement du suivi des cibles de s'adapter à votre cluster ElastiCache for Valkey et Redis OSS en désactivant l'activité de dimensionnement. La désactivation de l'activité de diminution en charge empêche la politique de mise à l'échelle de supprimer des réplicas, tout en l'autorisant encore à les créer si nécessaire.

Vous pouvez spécifier une valeur booléenne pour que `DisableScaleIn` active ou désactive l'activité de diminution en charge de votre cluster. Pour plus d'informations sur `DisableScaleIn`, consultez TargetTrackingScalingPolicyConfiguration dans le manuel Référence d'API Application Auto Scaling.

Example

L'exemple suivant décrit une configuration de suivi de la cible pour une politique de dimensionnement. Dans cette configuration, la métrique `ElastiCacheReplicaEngineCPUUtilization` prédéfinie ajuste un cluster en fonction d'une utilisation moyenne du processeur de 40 % pour toutes les répliques de ce cluster. La configuration désactive l'activité de diminution en charge pour la politique de dimensionnement.

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Appliquer une politique de dimensionnement à un cluster ElastiCache pour Valkey et Redis OSS

Après avoir enregistré votre cluster auprès ElastiCache de Valkey et Redis OSS Auto Scaling et défini une politique de dimensionnement, vous appliquez la politique de dimensionnement au cluster enregistré. Pour appliquer une politique de dimensionnement à un cluster ElastiCache pour Valkey et Redis OSS, vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling.

À l'aide du AWS CLI

Pour appliquer une politique de dimensionnement à votre cluster ElastiCache for Valkey et Redis OSS, utilisez la put-scaling-policycommande avec les paramètres suivants :

- --policy-name : nom de la politique de mise à l'échelle.

- --policy-type : définissez cette valeur à `TargetTrackingScaling`.
- --resource-id — Identifiant de ressource pour le cluster. Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace : définissez cette valeur à elasticache.
- --scalable-dimension : définit cette valeur à `elasticache:replication-group:Replicas`.
- -- target-tracking-scaling-policy -configuration — Configuration de la politique de dimensionnement du suivi des cibles à utiliser pour le cluster.

Example

Dans l'exemple suivant, vous appliquez une politique de dimensionnement de suivi des cibles nommée `myscalablepolicy` à un cluster nommé avec mise à l'échelle `myscalablecluster` ElastiCache automatique. Pour ce faire, vous utilisez une configuration de politique enregistrée dans un fichier nommé `config.json`.

Pour Linux, macOS ou Unix :

```
aws application-autoscaling put-scaling-policy \
    --policy-name myscalablepolicy \
    --policy-type TargetTrackingScaling \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:Replicas \
    --target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
    "DisableScaleIn": true
}
```

Pour Windows :

```
aws application-autoscaling put-scaling-policy ^
    --policy-name myscalablepolicy ^
```

```
    --policy-type TargetTrackingScaling ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:Replicas ^
    --target-tracking-scaling-policy-configuration file://config.json
```

Utilisation de l'API

Pour appliquer une politique de dimensionnement à votre ElastiCache cluster avec l'API Application Auto Scaling, utilisez l'opération PutScalingPolicyApplication Auto Scaling API avec les paramètres suivants :

- PolicyName — Le nom de la politique de dimensionnement.
- PolicyType — Définissez cette valeur sur`TargetTrackingScaling`.
- ResourceID — Identifiant de ressource pour le cluster. Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du ElastiCache cluster Redis OSS, par exemple`replication-group/myscalablecluster`.
- ServiceNamespace — Définissez cette valeur sur elasticache.
- ScalableDimension — Définissez cette valeur sur`elasticache:replication-group:Replicas`.
- TargetTrackingScalingPolicyConfiguration — La configuration de la politique de dimensionnement du suivi des cibles à utiliser pour le cluster.

Example

Dans l'exemple suivant, vous appliquez une politique de dimensionnement de suivi des cibles nommée `scalablepolicy` à un cluster nommé avec mise à l'échelle `myscalablecluster` ElastiCache automatique. Vous utilisez une configuration de politique basée sur la métrique prédéfinie `ElastiCacheReplicaEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
```

```
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas",
    "PolicyType": "TargetTrackingScaling",
    "TargetTrackingScalingPolicyConfiguration": {
        "TargetValue": 40.0,
        "PredefinedMetricSpecification":
        {
            "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
        }
    }
}
```

Modification d'une politique de dimensionnement

Vous pouvez modifier une politique de dimensionnement à l'aide de l' AWS Management Console API Application Auto Scaling ou de l'API Application Auto Scaling. AWS CLI

Modification d'une politique de dimensionnement à l'aide du AWS Management Console

Vous pouvez uniquement modifier des politiques avec le type métrique prédéfinies à l'aide de la AWS Management Console

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS
3. Choisissez le cluster auquel vous voulez ajouter une stratégie (choisissez le nom du cluster et non pas le bouton situé à sa gauche).
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Sous Scaling policies (Stratégies de dimensionnement), choisissez le bouton à gauche de la stratégie Auto Scaling que vous voulez changer, puis choisissez Modify (Modifier).
6. Apportez les modifications nécessaires à la politique.
7. Sélectionnez Modifier.
8. Apportez des modifications à la politique.
9. Sélectionnez Modifier.

Modification d'une politique de dimensionnement à l'aide de l'API Application Auto Scaling AWS CLI ou de l'API Application Auto Scaling

Vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling pour modifier une politique de dimensionnement de la même manière que vous appliquez une politique de dimensionnement :

- Lorsque vous utilisez l'API Application Auto Scaling, spécifiez le nom de la politique à modifier dans le paramètre `PolicyName`. Spécifiez de nouvelles valeurs pour les paramètres que vous souhaitez modifier.

Pour de plus amples informations, veuillez consulter Appliquer une politique de dimensionnement à un cluster ElastiCache pour Valkey et Redis OSS.

Suppression d'une politique de dimensionnement

Vous pouvez supprimer une politique de dimensionnement à l' AWS Management Console aide de l' AWS CLI API Application Auto Scaling

Suppression d'une politique de dimensionnement à l'aide du AWS Management Console

Vous pouvez uniquement modifier des politiques avec le type métrique prédéfinies à l'aide de la AWS Management Console

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS
3. Choisissez le cluster dont vous voulez supprimer la politique Auto Scaling.
4. Cliquez sur l'onglet Auto Scaling policies (Politiques de scalabilité automatique).
5. Sous Scaling policies (Stratégies de dimensionnement), choisissez la stratégie Auto Scaling, puis choisissez Delete (Supprimer).

Suppression d'une politique de dimensionnement à l'aide de l'API Application Auto Scaling AWS CLI ou de l'API Application Auto Scaling

Vous pouvez utiliser l'API Application Auto Scaling AWS CLI ou l'API Application Auto Scaling pour supprimer une politique de dimensionnement d'un ElastiCache cluster.

INTERFACE DE LIGNE DE COMMANDE (CLI)

Pour supprimer une politique de dimensionnement de votre cluster ElastiCache pour Valkey et Redis OSS, utilisez la delete-scaling-policycommande avec les paramètres suivants :

- --policy-name : nom de la politique de mise à l'échelle.
- --resource-id — Identifiant de ressource pour le cluster. Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- --service-namespace : définissez cette valeur à elasticache.
- --scalable-dimension – Définit cette valeur à `elasticache:replication-group:Replicas`.

Example

Dans l'exemple suivant, vous supprimez une politique de suivi de cible et d'échelonnement nommée `myscalablepolicy` d'un cluster ELC; nommé `myscalablecluster`.

Pour Linux, macOS ou Unix :

```
aws application-autoscaling delete-scaling-policy \
    --policy-name myscalablepolicy \
    --resource-id replication-group/myscalablecluster \
    --service-namespace elasticache \
    --scalable-dimension elasticache:replication-group:Replicas \
```

Pour Windows :

```
aws application-autoscaling delete-scaling-policy ^
    --policy-name myscalablepolicy ^
    --resource-id replication-group/myscalablecluster ^
    --service-namespace elasticache ^
    --scalable-dimension elasticache:replication-group:Replicas ^
```

API

Pour supprimer une politique de dimensionnement de votre cluster ElastiCache pour Valkey et Redis OSS, utilisez l'opération d'API DeleteScalingPolicyApplication Auto Scaling avec les paramètres suivants :

- PolicyName — Le nom de la politique de dimensionnement.

- ResourceID — Identifiant de ressource pour le cluster. Pour ce paramètre, le type de ressource est ReplicationGroup et l'identifiant unique est le nom du cluster, par exemple`replication-group/myscalablecluster`.
- ServiceNamespace — Définissez cette valeur sur elasticache.
- ScalableDimension — Définissez cette valeur sur`elasticache:replication-group:Replicas`.

Dans l'exemple suivant, vous supprimez une politique de dimensionnement de suivi des cibles nommée à `myscalablepolicy` partir d'un cluster nommé `myscalablecluster` avec l'API Application Auto Scaling.

```
POST / HTTP/1.1
>>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "PolicyName": "myscalablepolicy",
    "ServiceNamespace": "elasticache",
    "ResourceId": "replication-group/myscalablecluster",
    "ScalableDimension": "elasticache:replication-group:Replicas"
}
```

Utilisation AWS CloudFormation pour les politiques Auto Scaling

Cet extrait montre comment créer une action planifiée et l'appliquer à une AWS::ElastiCache::ReplicationGroupressource utilisant cette ressource. AWS::ApplicationAutoScaling::ScalableTarget Elle utilise les fonctions intrinsèques Fn::Join et Ref pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::ElastiCache::ReplicationGroup` qui est spécifiée dans le même modèle.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
```

```
      MaxCapacity: 0
      MinCapacity: 0
      ResourceId: !Sub replication-group/${logicalName}
      ScalableDimension: 'elasticache:replication-group:Replicas'
      ServiceNamespace: elasticache
      RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

  ScalingPolicy:
    Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
    Properties:
      ScalingTargetId: !Ref ScalingTarget
      ServiceNamespace: elasticache
      PolicyName: testpolicy
      PolicyType: TargetTrackingScaling
      ScalableDimension: 'elasticache:replication-group:Replicas'
      TargetTrackingScalingPolicyConfiguration:
        PredefinedMetricSpecification:
          PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
        TargetValue: 40
```

Mise à l'échelle planifiée

La mise à l'échelle en fonction d'une planification vous permet de mettre à l'échelle l'application en réponse aux changements de demande. Pour utiliser le dimensionnement planifié, vous créez des actions planifiées qui indiquent ElastiCache à Valkey et Redis OSS d'effectuer des activités de dimensionnement à des moments précis. Lorsque vous créez une action planifiée, vous spécifiez un ElastiCache cluster existant, le moment où l'activité de dimensionnement doit avoir lieu, la capacité minimale et la capacité maximale. Vous pouvez créer des actions planifiées pour une mise à l'échelle unique ou selon une planification récurrente.

Vous ne pouvez créer une action planifiée que pour les ElastiCache clusters déjà existants. Vous ne pouvez pas créer une action planifiée en même temps que vous créez un cluster.

Pour plus d'informations sur la terminologie relative à la création, à la gestion et à la suppression d'actions planifiées, veuillez consulter Commandes généralement utilisées pour la création, la gestion et la suppression d'actions planifiées

Pour créer une action planifiée unique :

Similaire à la dimension partition. Consultez Mise à l'échelle planifiée .

Pour supprimer une action planifiée

Similaire à la dimension partition. Consultez Mise à l'échelle planifiée .

Pour gérer la mise à l'échelle planifiée à l'aide de la AWS CLI

Utilisez la mise à l'échelle automatique des applications APIs suivante :

- put-scheduled-action
- describe-scheduled-actions
- delete-scheduled-action

AWS CloudFormation À utiliser pour créer des politiques Auto Scaling

Cet extrait montre comment créer une action planifiée et l'appliquer à une AWS::ElastiCache::ReplicationGroupressource utilisant cette ressource. AWS::ApplicationAutoScaling::ScalableTarget Elle utilise les fonctions intrinsèques Fn::Join et Ref pour construire la propriété `ResourceId` avec le nom logique de la ressource `AWS::ElastiCache::ReplicationGroup` qui est spécifiée dans le même modèle.

```
ScalingTarget:
   Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
   Properties:
     MaxCapacity: 0
     MinCapacity: 0
     ResourceId: !Sub replication-group/${logicalName}
     ScalableDimension: 'elasticache:replication-group:Replicas'
     ServiceNamespace: elasticache
     RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
     ScheduledActions:
       - EndTime: '2020-12-31T12:00:00.000Z'
         ScalableTargetAction:
           MaxCapacity: '5'
           MinCapacity: '2'
         ScheduledActionName: First
         Schedule: 'cron(0 18 * * ? *)'
```

Modification du mode cluster

Valkey et Redis OSS sont des bases de données en mémoire distribuées qui prennent en charge le sharding et la réplication. ElastiCache Les clusters Valkey et Redis OSS sont l'implémentation distribuée qui permet de partitionner les données sur plusieurs nœuds. Un cluster ElastiCache pour Redis OSS possède deux modes de fonctionnement, le mode cluster activé (CME) et le mode cluster désactivé (CMD). Dans CME, un moteur Valkey et Redis OSS fonctionne comme une base de données distribuée avec plusieurs partitions et nœuds, tandis que dans CMD, Valkey et Redis OSS fonctionnent comme un seul nœud.

Avant d'effectuer une migration de CMD à CME, les conditions suivantes doivent être remplies :

> **Important**
>
> La configuration du mode cluster ne peut être modifiée que du mode cluster désactivé vers le mode cluster activé. Il n'est pas possible d'inverser cette configuration.

- Le cluster ne peut avoir que des clés dans la base de données 0 uniquement.
- Les applications doivent utiliser un client Valkey ou Redis OSS capable d'utiliser le protocole Cluster et d'utiliser un point de terminaison de configuration.
- Le basculement automatique doit être activé sur le cluster avec au moins 1 réplica.
- La version minimale du moteur requise pour la migration est Valkey 7.2 et versions ultérieures, ou Redis OSS 7.0 et versions ultérieures.

Afin de migrer de CMD vers CME, la configuration du mode cluster doit être modifiée du mode cluster désactivé vers le mode cluster activé. Il s'agit d'une procédure en deux étapes qui garantit la disponibilité du cluster pendant le processus de migration.

> **Note**
>
> Vous devez fournir un groupe de paramètres avec une configuration de cluster activé, c'est-à-dire que le paramètre de cluster activé est défini sur `yes`. Si vous utilisez un groupe de paramètres par défaut, ElastiCache pour Redis, OSS choisira automatiquement le groupe de paramètres par défaut correspondant avec une configuration compatible avec les clusters. La valeur du paramètre de cluster activé est définie sur no pour un cluster CMD. Lorsque le

cluster passe en mode compatible, la valeur du paramètre de cluster activé est mise à jour vers `yes` dans le cadre de l'action de modification.
Pour de plus amples informations, consultez Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres.

1. Préparation : créez un cluster CME de test et assurez-vous que votre stack est prêt à fonctionner avec celui-ci. ElastiCache pour Redis OSS n'a aucun moyen de vérifier que vous êtes prêt. Pour de plus amples informations, veuillez consulter Création d'un cluster pour Valkey ou Redis OSS.
2. Modifier la configuration existante du cluster CMD pour qu'elle soit compatible avec le mode cluster — Dans ce mode, une seule partition sera déployée et, ElastiCache pour Redis OSS, elle fonctionnera comme un nœud unique mais également comme un cluster de partitions unique. Le mode compatible signifie que l'application cliente peut utiliser l'un ou l'autre des protocoles pour communiquer avec le cluster. Dans ce mode, les applications doivent être reconfigurées pour commencer à utiliser le protocole Valkey ou Redis OSS Cluster et le point de terminaison de configuration. Pour passer du mode cluster Valkey ou Redis OSS au mode cluster compatible, suivez les étapes ci-dessous :

   > **Note**
   >
   > En mode compatible, les autres opérations de modification telles que la mise à l'échelle et la version du moteur ne sont pas autorisées pour le cluster. De plus, les paramètres (à l'exception`cacheParameterGroupName`) ne peuvent pas être modifiés lors de la définition d'un paramètre en mode cluster dans la demande. ModifyReplicationGroup

   a. Utilisation du mode cluster AWS Management Console, consultez Modification d'un groupe de réplication et réglez le mode cluster sur Compatible
   b. À l'aide de l'API, consultez ModifyReplicationGroupet mettez à jour le `ClusterMode` paramètre sur`compatible`.
   c. À l'aide de AWS CLI, consultez modify-replication-groupet mettez à jour le `cluster-mode` paramètre sur`compatible`.

   Après avoir changé le mode de cluster Valkey ou Redis OSS en mode cluster compatible, l'DescribeReplicationGroupsAPI renverra le point de terminaison de configuration du ElastiCache cluster Redis OSS. Le point de terminaison de configuration du cluster est un point de

terminaison unique qui peut être utilisé par les applications pour se connecter au cluster. Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

3. Modifier la configuration du cluster en mode cluster activé : une fois que le mode cluster est défini comme compatible avec le mode cluster, la deuxième étape consiste à modifier la configuration du cluster en mode cluster activé. Dans ce mode, une seule partition est en cours d'exécution et les clients peuvent désormais mettre à l'échelle leurs clusters ou modifier d'autres configurations de cluster.

   Pour activer le mode cluster, procédez comme suit :

   Avant de commencer, assurez-vous que vos clients Valkey ou Redis OSS ont migré vers le protocole de cluster et que le point de terminaison de configuration du cluster n'est pas utilisé.

   a. À l'aide de AWS Management Console, consultez Modification d'un groupe de réplication et réglez le mode cluster sur Activé.
   b. À l'aide de l'API, consultez ModifyReplicationGroupet mettez à jour le `ClusterMode` paramètre sur`enabled`.
   c. À l'aide de AWS CLI, consultez modify-replication-groupet mettez à jour le `cluster-mode` paramètre sur`enabled`.

   Après avoir changé le mode cluster en mode activé, les points de terminaison seront configurés conformément à la spécification du cluster Valkey ou Redis OSS. L'DescribeReplicationGroupsAPI renverra le paramètre du mode cluster sous forme `enabled` de points de terminaison du cluster qui sont désormais disponibles pour être utilisés par les applications pour se connecter au cluster.

   Notez que les points de terminaison du cluster changent une fois que le mode cluster devient activé. Veillez à mettre à jour vos applications à l'aide des nouveaux points de terminaison.

Vous pouvez également choisir de revenir au mode cluster désactivé (CMD) à partir du mode cluster compatible et de conserver les configurations d'origine.

Modifier la configuration du cluster du mode cluster activé au mode cluster désactivé

1. Utilisation du mode cluster AWS Management Console, consultez Modification d'un groupe de réplication et réglez le mode cluster sur Disabled

2. À l'aide de l'API, consultez ModifyReplicationGroupet mettez à jour le `ClusterMode` paramètre sur`disabled`.
3. À l'aide de AWS CLI, consultez modify-replication-groupet mettez à jour le `cluster-mode` paramètre sur`disabled`.

Une fois le mode cluster désactivé, l'DescribeReplicationGroupsAPI renvoie le paramètre du mode cluster sous la forme`disabled`.

Réplication entre AWS régions à l'aide de banques de données mondiales

> **Note**
>
> Global Datastore n'est actuellement disponible que pour les clusters auto-conçus.

En utilisant la fonctionnalité Global Datastore, vous pouvez utiliser une réplication de clusters Valkey ou Redis OSS entièrement gérée, rapide, fiable et sécurisée dans toutes les régions. AWS Grâce à cette fonctionnalité, vous pouvez créer des clusters de répliques de lecture entre régions pour permettre des lectures à faible latence et une reprise après sinistre dans toutes les régions AWS .

Dans les sections suivantes, vous trouverez une description de l'utilisation des magasins de données globaux.

Rubriques

- Présentation
- Conditions préalables et limitations
- Utilisation des entrepôts de données globaux (console)
- Utilisation des entrepôts de données globaux (CLI)

Présentation

Chaque magasin de donnée global est un ensemble d'un ou de plusieurs clusters qui se répliquent les uns sur les autres.

Un magasin de données global se compose des éléments suivants :

- Cluster principal (actif) : un cluster principal accepte les écritures répliquées dans tous les clusters de l'entrepôt de données global. Un cluster principal accepte également les demandes de lecture.
- Cluster secondaire (passif) : un cluster secondaire accepte uniquement les demandes de lecture et réplique les mises à jour de données à partir d'un cluster principal. Un cluster secondaire doit se trouver dans une AWS région différente de celle du cluster principal.

Lorsque vous créez une banque de données globale ElastiCache pour Valkey ou Redis OSS, elle réplique automatiquement vos données du cluster principal vers le cluster secondaire. Vous choisissez la AWS région dans laquelle les données Valkey ou Redis OSS doivent être répliquées, puis vous créez un cluster secondaire dans cette région. AWS ElastiCache configure et gère ensuite la réplication automatique et asynchrone des données entre les deux clusters.

L'utilisation d'une banque de données globale pour Valkey ou Redis OSS présente les avantages suivants :

- Performances géolocalisées — En configurant des clusters de répliques distants dans des AWS régions supplémentaires et en synchronisant vos données entre elles, vous pouvez réduire la latence d'accès aux données dans cette région. AWS Une banque de données mondiale peut contribuer à améliorer la réactivité de votre application en proposant des lectures géolocalisées à faible latence dans toutes les régions. AWS
- Reprise après sinistre : si votre cluster principal d'un entrepôt de données global subit une dégradation, vous pouvez promouvoir un cluster secondaire en tant que nouveau cluster principal. Vous pouvez le faire en vous connectant à n'importe quelle AWS région contenant un cluster secondaire.

Le schéma suivant montre le fonctionnement des magasins de données globaux.

Conditions préalables et limitations

Avant de commencer à utiliser les magasins de données globaux, tenez compte des éléments suivants :

- Les banques de données mondiales sont prises en charge dans les régions suivantes : AWS
  - Afrique - Cape Town
  - Asie-Pacifique : Hong Kong, Hyderabad, Jakarta, Malaisie, Melbourne, Mumbai, Osaka, Séoul, Singapour, Sydney, Thaïlande et Tokyo
  - Canada - Centre du Canada et Canada Ouest (Calgary)
  - Chine - Pékin et Ningxia
  - Europe : Francfort, Londres, Irlande, Milan, Paris, Espagne, Stockholm et Zurich
  - AWS GovCloud-US-Ouest et US-Est
  - Israël - Tel Aviv
  - Moyen-Orient - Bahreïn et Émirats arabes unis
  - États-Unis - Est (Virginie du Nord et Ohio) et États-Unis Ouest (Californie du Nord et Oregon)
  - Amérique du Sud - Mexique (centre) et São Paulo
- Tous les clusters (primaire et secondaire) de votre entrepôt de données global doivent avoir le même nombre de nœuds principaux, type de nœud, version du moteur et nombre de partitions (dans le cas où le mode cluster est activé). Chaque cluster de votre magasin de données global

peut comporter un nombre différent de réplicas en lecture afin de tenir compte du trafic de lecture local vers ce cluster.

La réplication doit être activée si vous prévoyez d'utiliser un cluster à un seul nœud existant.

- Les banques de données globales sont prises en charge sur les instances de grande taille ou supérieure.
- Vous pouvez configurer la réplication d'un cluster principal d'une AWS région vers un cluster secondaire dans deux autres AWS régions au maximum.

> **Note**
>
> Les régions Chine (Beijing) et Chine (Ningxia), où la réplication ne peut se produire qu'entre les deux régions.

- Vous pouvez utiliser les magasins de données globaux uniquement dans des clusters VPC. Pour de plus amples informations, veuillez consulter Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC. Les banques de données globales ne sont pas prises en charge lorsque vous utilisez EC2 -Classic. Pour plus d'informations, consultez EC2-Classic dans le guide de l' EC2 utilisateur Amazon.

> **Note**
>
> Pour le moment, vous ne pouvez pas utiliser les entrepôts de données globaux dans Utilisation de zones locales avec ElastiCache.

- ElastiCache ne prend pas en charge le basculement automatique d'une AWS région à l'autre. Si nécessaire, vous pouvez promouvoir manuellement un cluster secondaire. Pour obtenir un exemple, consultez Promotion du cluster secondaire en cluster principal.
- Pour amorcer à partir de données existantes, utilisez un cluster existant en tant que cluster principal pour créer un magasin de données global. Nous ne prenons pas en charge l'ajout d'un cluster existant en tant que cluster secondaire. Le processus d'ajout d'un cluster existant en tant que cluster secondaire efface les données, ce qui peut entraîner une perte de données.
- Les mises à jour de paramètres sont appliquées à tous les clusters lorsque vous modifiez un groupe de paramètres local d'un cluster appartenant à un magasin de données global.
- Vous pouvez mettre à l'échelle les clusters régionaux à la fois verticalement (augmentation et diminution) et horizontalement (ajustement à la hausse et à la baisse). Vous pouvez mettre à l'échelle les clusters en modifiant le magasin de données global. Tous les clusters régionaux du

magasin de données global sont ensuite mis à l'échelle sans interruption. Pour de plus amples informations, veuillez consulter Dimensionnement ElastiCache.

- Les banques de données mondiales prennent en charge le chiffrement au repos, lechiffrement en transit et l'AUTH.
- Les banques de données globales ne prennent pas en charge le protocole Internet version 6 ()IPv6.
- Les banques de données mondiales prennent en charge les clés AWS KMS . Pour plus d'informations, veuillez consulter les Concepts du service de gestion des clés AWS dans le Guide du développeur AWS Key Management Service .

> **Note**
>
> Les magasins de données globaux prennent en charge la messagerie pub/sub (publish/subscribe - publier/abonner) avec les conditions suivantes :
>
> - Si le mode cluster est désactivé, il pub/sub est entièrement pris en charge. Les événements publiés sur le cluster principal de la AWS région principale sont propagés aux AWS régions secondaires.
> - Lorsque le mode cluster est activé, les conditions suivantes s'appliquent :
> - Pour les événements publiés qui ne figurent pas dans un keyspace, seuls les abonnés de la même AWS région reçoivent les événements.
> - Pour les événements keyspace publiés, les abonnés de toutes les AWS régions reçoivent les événements.

Utilisation des entrepôts de données globaux (console)

Pour créer un magasin de données global à l'aide de la console, suivez ce processus en deux étapes :

1. Créez un cluster principal, soit en utilisant un cluster existant, soit en créant un nouveau cluster. Le moteur doit être Valkey 7.2 ou version ultérieure, ou Redis OSS 5.0.6 ou version ultérieure.
2. Ajoutez jusqu'à deux clusters secondaires dans différentes AWS régions, toujours en utilisant Valkey 7.2 ou version ultérieure, ou Redis OSS 5.0.6 ou version ultérieure.

Les procédures suivantes vous expliquent comment créer une banque de données globale pour Valkey ou Redis OSS et comment effectuer d'autres opérations à l'aide de la console. ElastiCache

Rubriques

- Création d'un entrepôt de données global à l'aide d'un cluster existant
- Création d'un entrepôt de données global à l'aide d'un nouveau cluster principal
- Affichage des détails de l'entrepôt de données global
- Ajout d'une région à un entrepôt de données global
- Modification d'un entrepôt de données global
- Promotion du cluster secondaire en cluster principal
- Suppression d'une région d'un entrepôt de données global
- Suppression d'un entrepôt de données global

Création d'un entrepôt de données global à l'aide d'un cluster existant

Dans ce scénario, vous utilisez un cluster existant comme cluster principal du nouveau magasin de données global. Vous créez ensuite un cluster secondaire, en lecture seule, dans une autre région AWS . Ce cluster secondaire reçoit des mises à jour automatiques et asynchrones du cluster principal.

> **Important**
>
> Le cluster existant doit utiliser un moteur Valkey 7.2 ou version ultérieure ou Redis OSS 5.0.6 ou version ultérieure.

Pour créer un magasin de données global à l'aide d'un cluster existant

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Global Datastores, puis Create global datastore.
3. Sur la page des paramètres du cluster principal, procédez comme suit :
   - Dans le champ Informations sur la banque de données globale, entrez le nom de la nouvelle banque de données globale.
   - (Facultatif) Entrez une valeur dans le champ Description.

4. Sous Cluster régional, sélectionnez Utiliser le cluster régional existant.
5. Sous Cluster existant, sélectionnez le cluster existant que vous souhaitez utiliser.
6. Ne modifiez pas les options suivantes. Elles sont prérenseignées pour correspondre à la configuration du cluster principal. Vous ne pouvez pas les modifier.

   - Version de moteur
   - Type de nœud
   - Groupe de paramètres

     > **Note**
     >
     > ElastiCache génère automatiquement un nouveau groupe de paramètres à partir des valeurs du groupe de paramètres fourni et applique le nouveau groupe de paramètres au cluster. Utilisez ce nouveau groupe de paramètres pour modifier les paramètres d'un entrepôt de données global. Chaque groupe de paramètres généré automatiquement est associé à un seul cluster et, par conséquent, à un seul entrepôt de données global.

   - Nombre de partitions
   - Chiffrement au repos : active le chiffrement des données stockées sur le disque. Pour de plus amples informations, veuillez consulter Chiffrement au repos.

     > **Note**
     >
     > Vous pouvez fournir une autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour plus d'informations, consultez la section Utilisation de clés AWS KMS gérées par le client.

   - Chiffrement en transit : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour Valkey 7.2 et versions ultérieures et Redis OSS 6.0 et versions ultérieures, si vous activez le chiffrement en transit, vous êtes invité à spécifier l'une des options de contrôle d'accès suivantes :
     - Aucun contrôle d'accès : il s'agit du paramètre par défaut. Cela indique qu'il n'y a pas de restrictions.
     - Liste de contrôle d'accès aux groupes d'utilisateurs : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs et d'autorisations sur les opérations disponibles. Pour

de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.

- Utilisateur par défaut AUTH : mécanisme d'authentification pour un serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.

7. (Facultatif) Si besoin, mettez à jour les autres paramètres des clusters secondaires. Ils sont prérenseignés avec les mêmes valeurs que celles du cluster principal, mais vous pouvez les mettre à jour pour répondre à des exigences spécifiques pour ce cluster.

   - Port
   - Nombre de réplicas
   - Groupe de sous-réseaux
   - Zone(s) de disponibilité préférée(s)
   - Groupes de sécurité
   - Géré par le client (cléAWS KMS)
   - Jeton AUTH
   - Activer les sauvegardes automatiques
   - Période de rétention des sauvegardes
   - Fenêtre de sauvegarde
   - Fenêtre de maintenance
   - Rubrique pour la notification SNS

8. Choisissez Créer. Cette opération définit l'état du magasin de données global sur Creating (En cours de création). L'état passe à Modifying (En cours de modification) après association du cluster principal au magasin de données global et passage du cluster secondaire à l'état Associating (En cours d'association) .

   Une fois que le cluster principal et les clusters secondaires sont associés au magasin de données global, l'état passe à Available (Disponible). À ce stade, vous disposez d'un cluster principal qui accepte les lectures et les écritures, et de clusters secondaires qui acceptent les lectures répliquées à partir du cluster principal.

   La page est mise à jour pour indiquer si un cluster fait partie d'une banque de données globale, notamment :

   - Global Datastore (Magasin de données global) : nom de l'entrepôt de données global auquel appartient le cluster.

- Global Datastore Role (Rôle de l'entrepôt de données global) : rôle du cluster, principal ou secondaire.

Vous pouvez ajouter jusqu'à un cluster secondaire supplémentaire dans une AWS région différente. Pour de plus amples informations, veuillez consulter Ajout d'une région à un entrepôt de données global.

Création d'un entrepôt de données global à l'aide d'un nouveau cluster principal

Si vous choisissez de créer un entrepôt de données global avec un nouveau cluster, procédez comme suit.

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Global Datastores, puis Create global datastore.
3. Sous Primary cluster settings (Paramètres de cluster principal), procédez comme suit :
   a. Pour Cluster mode (Mode du cluster), choisissez Enabled (Activé) ou Disabled (Désactivé).
   b. Pour les informations sur la banque de données globale, entrez une valeur pour le nom. ElastiCache utilise le suffixe pour générer un nom unique pour la banque de données globale. Vous pouvez rechercher l'entrepôt de données global à l'aide du suffixe que vous spécifiez ici.
   c. (Facultatif) Entrez une valeur pour Global Datastore Description (Description du magasin de données global).
4. Sous Regional cluster (Cluster régional) :
   a. Pour Région, choisissez une AWS région disponible.
   b. Choisissez Create new regional cluster (Créer un nouveau cluster régional) ou Use existing regional cluster (Utiliser un cluster régional existant)
   c. Si vous choisissez Create new regional cluster (Créer un nouveau cluster régional), sous Cluster info (Infos sur le cluster), saisissez un nom et une description facultative du cluster.
   d. Sous Location (Emplacement), nous vous recommandons d'accepter les paramètres par défaut pour Multi-AZ et Auto-failover (Basculement automatique).
5. Sous Cluster settings (Paramètres de cluster)

a. Pour Engine version (Version du moteur), choisissez une version disponible, à savoir 5.0.6 ou ultérieure.

b. Pour Port, utilisez le port par défaut, 6379. Si vous avez une raison d'utiliser un autre port, saisissez le numéro de port.

c. Pour Groupe de paramètres, choisissez un groupe de paramètres ou créez-en un nouveau. Les groupes de paramètres contrôlent les paramètres d'exécution de votre cluster. Pour plus d'informations sur les groupes de paramètres, consultez Paramètres OSS Valkey et Redis et Création d'un groupe ElastiCache de paramètres.

> **Note**
>
> Lorsque vous sélectionnez un groupe de paramètres pour définir les valeurs de configuration du moteur, ce groupe de paramètres est appliqué à tous les clusters du magasin de données global. Dans la page Parameter Groups (Groupes de paramètres) l'attribut Global yes/no (oui/non) indique si un groupe de paramètres fait partie d'un magasin de données global.

d. Pour Type de nœud, choisissez la flèche vers le bas
(▾).
Dans la boîte de dialogue Modifier le type de nœud choisissez une valeur pour la famille d'instances pour le type de nœud souhaité. Choisissez ensuite le type de nœud que vous souhaitez utiliser pour ce cluster, puis choisissez Enregistrer.

   Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

   Si vous choisissez un type de nœud r6gd, la hiérarchisation des données est automatiquement activée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

e. Si vous créez un cluster Valkey ou Redis OSS (mode cluster désactivé) :

   Pour Number of replicas (Nombre de réplicas), choisissez le nombre de réplicas que vous voulez pour ce cluster.

f. Si vous créez un cluster Valkey ou Redis OSS (mode cluster activé) :

   i. Pour Nombre de partitions, choisissez le nombre de partitions (partitions/groupes de nœuds) que vous souhaitez pour ce cluster Valkey ou Redis OSS (mode cluster activé).

Pour certaines versions de Valkey ou Redis OSS (mode cluster activé), vous pouvez modifier dynamiquement le nombre de partitions de votre cluster :

- Redis OSS 3.2.10 et versions ultérieures : si votre cluster exécute Redis OSS 3.2.10 ou versions ultérieures, vous pouvez modifier le nombre de partitions de votre cluster de manière dynamique. Pour de plus amples informations, veuillez consulter Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).
- Autres versions de Redis OSS — Si votre cluster exécute une version de Redis OSS antérieure à la version 3.2.10, il existe une autre approche. Pour modifier le nombre de fragments dans votre cluster dans ce cas, créez un nouveau cluster avec le nouveau nombre de fragments. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.

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

Les restrictions suivantes existent pour Valkey ou Redis OSS (mode cluster activé).

- 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.
- Le nombre de réplicas de lecture par partition est fixe et ne peut pas être modifié. Si vous avez besoin de davantage ou de moins de réplicas par partition (API/CLI : groupe de nœuds), vous devez créer un cluster avec le nouveau nombre de réplicas. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un nouveau cluster conçu par vos soins avec une sauvegarde créée en externe.

6. Pour les paramètres du groupe de sous-réseaux, choisissez le sous-réseau que vous souhaitez appliquer à ce cluster. ElastiCache fournit un groupe de IPv4 sous-réseaux par défaut ou vous pouvez choisir d'en créer un nouveau. En IPv6 effet, vous devez créer un groupe de sous-réseaux avec un bloc IPv6 CIDR. Si vous choisissez la double pile, vous devez sélectionner un type d'IP de découverte, IPv6 soit IPv4.

   Pour plus d'informations, consultez Création d'un sous-réseau dans votre VPC.

7. Pour Availability zone placements (Placement de zones de disponibilité), vous avez deux options :

   - Aucune préférence : ElastiCache choisit la zone de disponibilité.

- Specify availability zones (Spécifier les zones de disponibilité) : vous spécifiez la zone de disponibilité pour chaque cluster.

  Si vous avez choisi de spécifier les Zones de disponibilité, pour chaque cluster de chaque partition, choisissez la Zone de disponibilité depuis la liste.

Pour de plus amples informations, veuillez consulter Choix des régions et des zones de disponibilité pour ElastiCache.

Définition des Keyspaces et des zones de disponibilité

8. Choisissez Next (Suivant)
9. Dans les paramètres avancés de Valkey et Redis OSS

   - Pour Security (Sécurité) :

     i. 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 autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour de plus amples informations, veuillez consulter Utilisation clés AWS KMS gérées par le client.

- Encryption in-transit (Chiffrement en transit) : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour Valkey 7.2 et versions ultérieures et Redis OSS 6.0 et versions ultérieures, si vous activez le chiffrement en transit, vous serez invité à spécifier l'une des options de contrôle d'accès suivantes :
  - No Access Control (Aucun contrôle d'accès) – il s'agit du paramètre par défaut. Cela indique qu'aucune restriction n'est imposée à l'accès des utilisateurs au cluster.
  - User Group Access Control List (Liste de contrôle d'accès au groupe d'utilisateurs) : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs pouvant accéder au cluster. Pour de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.
  - Utilisateur par défaut AUTH : mécanisme d'authentification pour un serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.
- AUTH — Mécanisme d'authentification pour un serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.

> **Note**
>
> Pour les versions de Redis OSS supérieures à 3.2.6, à l'exception de la version 3.2.10, AUTH est la seule option.

ii. 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.

10. Pour des sauvegardes automatiques régulières, choisissez Activer les sauvegardes automatiques, puis entrez le nombre de jours pendant lesquels vous souhaitez conserver une sauvegarde automatique avant sa suppression automatique. Si vous ne souhaitez pas de sauvegardes automatiques régulières, désactivez la case à cocher Enable automatic backups. Dans les deux cas, vous avez toujours la possibilité de créer des sauvegardes manuelles.

    Pour plus d'informations sur la sauvegarde et la restauration, consultezInstantané et restauration.

11. (Facultatif) Spécifiez une fenêtre de maintenance. La fenêtre de maintenance indique le moment, généralement d'une durée d'une heure, chaque semaine, où ElastiCache planifie la maintenance du système pour votre cluster. Vous pouvez ElastiCache autoriser le choix du jour et de 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 de plus amples informations, veuillez consulter Gestion de la maintenance des ElastiCache clusters.

12. (Facultatif) Pour Logs (Journaux) :

    - Sous Log format (Format de journal), sélectionnez Text (Texte) ou JSON.
    - Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
    - Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatch journaux ou le nom de votre flux Firehose, ou choisissez Select existing, puis choisissez le nom de votre groupe de journaux de CloudWatch journaux ou le nom de votre stream Firehose,

13. Pour les balises, pour vous aider à gérer vos clusters et autres ElastiCache ressources, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Pour plus d'informations, consultez Marquer vos ressources ElastiCache .

14. Passez en revue toutes vos entrées et sélections, puis effectuez les corrections nécessaires. Lorsque vous avez terminé, choisissez Next (Suivant).

15. Une fois que vous avez configuré le cluster dans les étapes précédentes, vous configurez maintenant les détails de votre cluster secondaire.

16. Sous Cluster régional, choisissez la AWS région où se trouve le cluster.
17. Sous Cluster info (Infos sur le cluster), saisissez un nom et une description facultative du cluster.
18. Les options suivantes sont prérenseignées pour correspondre à la configuration du cluster principal et ne peuvent pas être modifiés :

    - Emplacement
    - Version de moteur
    - Type d'instance
    - Type de nœud
    - Nombre de partitions
    - Groupe de paramètres

    > **Note**
    >
    > ElastiCache génère automatiquement un nouveau groupe de paramètres à partir des valeurs du groupe de paramètres fourni et applique le nouveau groupe de paramètres au cluster. Utilisez ce nouveau groupe de paramètres pour modifier les paramètres d'un entrepôt de données global. Chaque groupe de paramètres généré automatiquement est associé à un seul cluster et, par conséquent, à un seul entrepôt de données global.

    - 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 pouvez fournir une autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé. Pour plus d'informations, consultez la section Utilisation de clés AWS KMS gérées par le client.

    - Chiffrement en transit : permet le chiffrement des données sur le câble. Pour de plus amples informations, veuillez consulter Chiffrement en transit. Pour Valkey 7.2 et versions ultérieures et Redis OSS 6.4 et versions ultérieures, si vous activez le chiffrement en transit, vous êtes invité à spécifier l'une des options de contrôle d'accès suivantes :
        - No Access Control (Aucun contrôle d'accès) – il s'agit du paramètre par défaut. Cela indique qu'aucune restriction n'est imposée à l'accès des utilisateurs au cluster.

- User Group Access Control List (Liste de contrôle d'accès au groupe d'utilisateurs) : choisissez un groupe d'utilisateurs avec un ensemble défini d'utilisateurs pouvant accéder au cluster. Pour de plus amples informations, veuillez consulter Gestion des groupes d'utilisateurs avec la console et la CLI.
- Utilisateur par défaut AUTH : mécanisme d'authentification pour un serveur Valkey ou Redis OSS. Pour plus d'informations, consultez AUTH.

> **Note**
>
> Pour les versions de Redis OSS comprises entre 4.0.2, lorsque le chiffrement en transit a été pris en charge pour la première fois, et 6.0.4, AUTH est la seule option.

Les autres paramètres de cluster secondaire sont prérenseignés avec les mêmes valeurs que celles du cluster principal, mais les éléments suivants peuvent être mis à jour pour répondre à aux exigences spécifiques de ce cluster :

- Port
- Nombre de réplicas
- Groupe de sous-réseaux
- Zone(s) de disponibilité préférée(s)
- Groupes de sécurité
- Géré par le client (cléAWS KMS)
- Jeton AUTH
- Activer les sauvegardes automatiques
- Période de rétention des sauvegardes
- Fenêtre de sauvegarde
- Fenêtre de maintenance
- Rubrique pour la notification SNS

19. Choisissez Créer. Cette opération définit l'état du magasin de données global sur Creating (En cours de création). Une fois que le cluster principal et les clusters secondaires sont associés au magasin de données global, l'état passe à Available (Disponible). Vous disposez d'un cluster principal qui accepte les lectures et les écritures, et d'un cluster secondaire qui accepte les lectures répliquées à partir du cluster principal.

La page est également mise à jour pour indiquer si un cluster fait partie d'une banque de données globale, notamment les éléments suivants :

- Global Datastore (Magasin de données global) : nom de l'entrepôt de données global auquel appartient le cluster.
- Global Datastore Role (Rôle de l'entrepôt de données global) : rôle du cluster, principal ou secondaire.

Vous pouvez ajouter jusqu'à un cluster secondaire supplémentaire dans une AWS région différente. Pour de plus amples informations, veuillez consulter Ajout d'une région à un entrepôt de données global.

Affichage des détails de l'entrepôt de données global

Vous pouvez consulter les détails des banques de données globales existantes et également les modifier sur la page Banques de données globales.

Pour afficher les détails du magasin de données global

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Global Datastores, puis choisissez une banque de données globale disponible.

Vous pouvez ensuite examiner les propriétés du magasin de données global suivantes :

- Global Datastore Name (Nom du magasin de données global) : Nom du magasin de données global
- Description : Description du magasin de données global
- Status (État) : Les options comprennent :
  - Création
  - Modification
  - Disponible
  - Suppression

  - Primary Only (Principal uniquement) - Cet état indique que le magasin de données global ne contient qu'un cluster principal. Tous les clusters secondaires sont supprimés ou leur création a échoué.
- Cluster Mode (Mode cluster) : Activé ou désactivé
- Version du moteur : version du moteur Valkey ou Redis OSS exécutant la banque de données globale
- Instance Node Type (Type de nœud d'instance) : Type de nœud utilisé pour le magasin de données global
- Encryption at-rest (Chiffrement au repos) : Activé ou désactivé
- Encryption in-transit (Chiffrement en transit) : Activé ou désactivé
- AUTH : activé ou désactivé

Vous pouvez apporter les modifications suivantes au magasin de données global :

- Ajout d'une région à un entrepôt de données global
- Suppression d'une région d'un entrepôt de données global
- Promotion du cluster secondaire en cluster principal
- Modification d'un entrepôt de données global

La page Global Datastore (Magasin de données global) répertorie également les clusters individuels qui composent le magasin de données global et les propriétés suivantes pour chacun d'eux :

- Région : AWS région dans laquelle le cluster est stocké
- Role (Rôle) - Principal ou secondaire
- Cluster name (Nom du cluster) - Nom du cluster
- Statut (État) - Les options comprennent :
  - Associating (En cours d'association) - Le cluster est en cours d'association au magasin de données global
  - Associated (Associé) - Le cluster est associé au magasin de données global
  - Disassociating (En cours de dissociation) - Processus de suppression d'un cluster secondaire du magasin de données global à l'aide du nom de ce dernier. Ensuite, le cluster secondaire ne reçoit plus de mises à jour du cluster principal, mais il reste un cluster autonome dans cette AWS région.

- Disassociated (Dissocié) - Le cluster secondaire a été supprimé de l'entrepôt de données global et est désormais un cluster autonome dans sa région AWS .
- Retard global de réplication de la banque de données : affiche une valeur par AWS région secondaire dans la banque de données globale. Il s'agit du décalage entre le nœud primaire de la région secondaire et le nœud primaire de la région primaire. Pour Valkey ou Redis OSS activé en mode cluster, le décalage indique le délai maximal en secondes entre les partitions.

Ajout d'une région à un entrepôt de données global

Vous pouvez ajouter jusqu'à une AWS région supplémentaire à une banque de données mondiale existante. Dans ce scénario, vous créez un cluster en lecture seule dans une AWS région distincte qui reçoit des mises à jour automatiques et asynchrones du cluster principal.

Pour ajouter une AWS région à une banque de données globale

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Global Datastores, puis sélectionnez une banque de données globale existante.
3. Choisissez Ajouter un cluster régional, puis choisissez la AWS région dans laquelle le cluster secondaire doit résider.
4. Sous Informations sur le cluster, entrez une valeur pour le nom et, éventuellement, pour la description du cluster.
5. Ne modifiez pas les options suivantes. Elles sont prérenseignées pour correspondre à la configuration du cluster principal. Vous ne pouvez pas les modifier.

   - Version de moteur
   - Type d'instance
   - Type de nœud
   - Nombre de partitions
   - Groupe de paramètres

   > **Note**
   >
   > ElastiCache génère automatiquement un nouveau groupe de paramètres à partir des valeurs du groupe de paramètres fourni et applique le nouveau groupe de

paramètres au cluster. Utilisez ce nouveau groupe de paramètres pour modifier les paramètres d'un entrepôt de données global. Chaque groupe de paramètres généré automatiquement est associé à un seul cluster et, par conséquent, à un seul entrepôt de données global.

- Chiffrement au repos

> **Note**
>
> Vous pouvez fournir une autre clé de chiffrement en choisissant la clé AWS KMS gérée par le client et en choisissant la clé.

- Chiffrement en transit
- AUTH

6. (Facultatif) Mettez à jour les autres paramètres des clusters secondaires. Ils sont prérenseignés avec les mêmes valeurs que celles du cluster principal, mais vous pouvez les mettre à jour pour répondre à des exigences spécifiques pour ce cluster :

   - Port
   - Nombre de réplicas
   - Groupe de sous-réseaux
   - Zone(s) de disponibilité préférée(s)
   - Groupes de sécurité
   - AWS KMS géré par le client (clé)
   - Jeton AUTH
   - Activer les sauvegardes automatiques
   - Période de rétention des sauvegardes
   - Fenêtre de sauvegarde
   - Fenêtre de maintenance
   - Rubrique pour la notification SNS
7. Choisissez Ajouter.

Modification d'un entrepôt de données global

Vous pouvez modifier les propriétés des clusters régionaux. Une seule opération de modification peut être en cours sur un magasin de données global, à l'exception de la promotion d'un cluster secondaire en cluster principal. Pour de plus amples informations, veuillez consulter Promotion du cluster secondaire en cluster principal.

Pour modifier un magasin de données global

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Global Datastores, puis pour Global Datastore Name, choisissez une banque de données globale.
3. Choisissez Modify (Modifier) et choisissez l'une des options suivantes :

   - Modify description (Modifier la description) : mettre à jour la description de l'entrepôt de données global
   - Modifier la version du moteur — Seuls Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.6 et versions ultérieures sont disponibles.
   - Modify node type (Modifier le type de nœud) : mettre à l'échelle les clusters régionaux à la fois verticalement (augmentation et diminution du nombre de ressources) et horizontalement (redimensionnement à la hausse et à la baisse du nombre de noeuds). Les options incluent les familles de nœuds R5 et M5. Pour de plus amples informations sur les types de nœud, veuillez consulter Types de nœuds pris en charge.
   - Modification du basculement automatique : activez ou désactivez le basculement automatique. Lorsque vous activez le basculement et que les nœuds principaux des clusters régionaux s'arrêtent de manière inattendue, ils ElastiCache basculent vers l'une des répliques régionales. Pour de plus amples informations, veuillez consulter Basculement automatique.

   Pour les clusters Valkey ou Redis OSS avec le mode cluster activé :

   - Add shards (Ajouter des partitions) : entrez le nombre de partitions à ajouter et spécifiez éventuellement une ou plusieurs zones de disponibilité.
   - Supprimer les partitions : choisissez les partitions à supprimer dans chaque AWS région.
   - Rebalance shards (Rééquilibrer les partitions) : rééquilibrez la distribution des emplacements pour assurer une distribution uniforme entre les partitions existantes dans le cluster.

Pour modifier les paramètres d'une banque de données globale, modifiez le groupe de paramètres de n'importe quel cluster membre de la banque de données globale. ElastiCache applique automatiquement cette modification à tous les clusters de cette banque de données globale.
Pour modifier le groupe de paramètres de ce cluster, utilisez la console Valkey ou Redis OSS ou l'opération ModifyCacheClusterAPI. Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres. Lorsque vous modifiez le groupe de paramètres d'un cluster contenu dans un magasin de données global, la modification est appliquée à tous les clusters dans ce magasin de données global.

Pour réinitialiser un groupe de paramètres entier ou des paramètres spécifiques, utilisez l'opération ResetCacheParameterGroupAPI.

Promotion du cluster secondaire en cluster principal

Si le cluster ou la AWS région principal devient indisponible ou rencontre des problèmes de performances, vous pouvez transformer un cluster secondaire en cluster principal. La promotion est autorisée à tout moment, même si d'autres modifications sont en cours. Vous pouvez également effectuer plusieurs promotions en parallèle. Au final, le magasin de données global contiendra un seul cluster principal. Si vous promouvez plusieurs clusters secondaires simultanément, ElastiCache cela ne garantit pas lequel sera finalement transformé en cluster principal.

Pour promouvoir un cluster secondaire en cluster principal

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, sélectionnez Global Datastores.
3. Sélectionnez le nom de l'entrepôt de données global pour en afficher les détails.
4. Choisissez le cluster Secondary (Secondaire).
5. Choisissez Promote to primary (Promouvoir en cluster principal).

   L'avertissement suivant vous invite à confirmer votre décision : `Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the secondary cluster to primary?`

   `The current primary cluster in primary region will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.`

6. Choisissez Confirm (Confirmer) si vous souhaitez continuer la promotion ou Cancel (Annuler) si vous ne le souhaitez pas.

Si vous choisissez de confirmer, votre magasin de données global passe à l'état Modifying (En cours de modification) et n'est pas disponible tant que la promotion n'est pas terminée.

Suppression d'une région d'un entrepôt de données global

Vous pouvez supprimer une AWS région d'une banque de données globale à l'aide de la procédure suivante.

Pour supprimer une AWS région d'une banque de données globale

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, sélectionnez Global Datastores.
3. Sélectionnez un entrepôt de données global.
4. Choisissez la région que vous souhaitez supprimer.
5. Choisissez Remove region (Supprimer la région).

> **Note**
> Cette option n'est disponible que pour les clusters secondaires.

L'avertissement suivant vous invite à confirmer votre décision : `Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?`

6. Choisissez Confirm (Confirmer) si vous souhaitez continuer la promotion ou Cancel (Annuler) si vous ne le souhaitez pas.

Si vous choisissez Confirmer, la AWS région est supprimée et le cluster secondaire ne reçoit plus de mises à jour de réplication.

Suppression d'un entrepôt de données global

Pour supprimer un magasin de données global, commencez par supprimer tous les clusters secondaires. Pour de plus amples informations, veuillez consulter Suppression d'une région d'un entrepôt de données global. L'état du magasin de données global reste alors primary-only (principal seulement).

Pour supprimer un magasin de données global

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, sélectionnez Global Datastores.
3. Sous Global Datastore Name (Nom du magasin de données global) de données globale, choisissez le magasin de données global à supprimer, puis choisissez Delete (Supprimer).

   L'avertissement suivant vous invite à confirmer votre décision : `Are you sure you want to delete this Global Datastore?`
4. Sélectionnez Delete (Supprimer).

Le magasin de données global passe à l'état Deleting (En cours de suppression) .

Utilisation des entrepôts de données globaux (CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour contrôler plusieurs AWS services à partir de la ligne de commande et les automatiser par le biais de scripts. Vous pouvez utiliser la AWS CLI pour des opérations ad hoc (ponctuelles).

Téléchargement et configuration du AWS CLI

Il AWS CLI fonctionne sous Windows, macOS ou Linux. Suivez la procédure suivante pour la télécharger et la configurer.

Pour télécharger, installer et configurer la CLI

1. Téléchargez la AWS CLI sur la page Web de l'interface de ligne de AWS commande.
2. Suivez les instructions d'installation de la AWS CLI et de configuration de la AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Utilisation de la AWS CLI avec des banques de données globales

Utilisez les opérations de la CLI suivantes pour utiliser les magasins de données globaux :

- create-global-replication-group

```
aws elasticache create-global-replication-group \
   --global-replication-group-id-suffix my global datastore  \
   --primary-replication-group-id sample-repl-group  \
   --global-replication-group-description an optional description of the global
 datastore
```

Amazon applique ElastiCache automatiquement un préfixe à l'identifiant global de la banque de données lors de sa création. Chaque AWS région possède son propre préfixe. Par exemple, un ID d'entrepôt de données global créé dans la région USA Ouest (Californie du Nord) commence par « virxk » avec le nom de suffixe que vous fournissez. Le suffixe, combiné au préfixe autogénéré, garantit l'unicité du nom de l'entrepôt de données global dans plusieurs régions.

Le tableau suivant répertorie chaque AWS région et son préfixe d'ID de banque de données global.

Nom de région/Région	Préfixe
Région US East (Ohio) `us-east-2`	`fpkhr`
Région US East (N. Virginia) `us-east-1`	`ldgnf`
Région US West (N. California) `us-west-1`	`virxk`
Région US West (Oregon) `us-west-2`	`sgaui`
Région Canada (Centre)	`bxodz`

Nom de région/Région	Préfixe
`ca-central-1`	
Région Asia Pacific (Mumbai) `ap-south-1`	`erpgt`
Région Asia Pacific (Tokyo) `ap-northeast-1`	`quwsw`
Région Asia Pacific (Seoul) `ap-northeast-2`	`lfqnh`
Région Asie-Pacifique (Osaka) `ap-northeast-3`	`nlapn`
Région Asia Pacific (Singapore) `ap-southeast-1`	`vlqxn`
Région Asia Pacific (Sydney) `ap-southeast-2`	`vbgxd`
Région Europe (Frankfurt) `eu-central-1`	`iudkw`
Région Europe (Irlande) `eu-west-1`	`gxeiz`
Région Europe (London) `eu-west-2`	`okuqm`

Nom de région/Région	Préfixe
Région Europe (Paris) `eu-west-3`	`fgjhi`
Région South America (São Paulo) `sa-east-1`	`juxlw`
Région Chine (Beijing) `cn-north-1`	`emvgo`
Région Chine (Ningxia) `cn-northwest-1`	`ckbem`
Région Asie-Pacifique (Hong Kong) `ap-east-1`	`knjmp`
AWS GovCloud (US-Ouest) `us-gov-west-1`	`sgwui`

- create-replication-group— Utilisez cette opération pour créer des clusters secondaires pour une banque de données globale en fournissant le nom de la banque de données globale au `--global-replication-group-id` paramètre.

```
aws elasticache create-replication-group \
  --replication-group-id secondary replication group name \
  --replication-group-description “Replication group description" \
  --global-replication-group-id global datastore name
```

Lorsque vous appelez cette opération et que vous `--global-replication-group-id` transmettez une valeur, les valeurs ElastiCache seront déduites du groupe de réplication principal du groupe de réplication global pour les paramètres suivants. Ne transmettez pas de valeurs pour ces paramètres :

```
"PrimaryClusterId",

"AutomaticFailoverEnabled",

"NumNodeGroups",

"CacheParameterGroupName",

"CacheNodeType",

"Engine",

"EngineVersion",

"CacheSecurityGroupNames",

"EnableTransitEncryption",

"AtRestEncryptionEnabled",

"SnapshotArns",

"SnapshotName"
```

- describe-global-replication-groups

```
aws elasticache describe-global-replication-groups \
   --global-replication-group-id my global datastore  \
   --show-member-info an optional parameter that returns a list of the primary and
 secondary clusters that make up the global datastore
```

- modify-global-replication-group

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id my global datastore  \
   --automatic-failover-enabled \
   --cache-node-type node type \
   --cache-parameter-group-name parameter group name \
   --engine-version engine version \
   --—apply-immediately \
   --global-replication-group-description description
```

Mise à niveau multimoteur de Redis vers OSS Valkey pour ElastiCache GlobalDataStore

Vous pouvez mettre à niveau un groupe de réplication global Redis OSS existant vers Valkey à l'aide de la console, de l'API ou de la CLI.

Si vous disposez d'un groupe de réplication global Redis OSS existant, vous pouvez passer à Valkey en spécifiant le nouveau moteur et la nouvelle version du moteur avec modify-global-replication-group l'API.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id myGlobalReplGroup \
   --engine valkey \
   --apply-immediately \
   --engine-version 8.0
```

Pour Windows :

```
aws elasticache modify-global-replication-group ^
   --global-replication-group-id myGlobalReplGroup ^
   --engine valkey ^
   --apply-immediately ^
   --engine-version 8.0
```

Si un groupe de paramètres de cache personnalisé est appliqué au groupe de réplication global Redis OSS existant que vous souhaitez mettre à niveau, vous devrez également transmettre un groupe de paramètres de cache Valkey personnalisé dans la demande. Le groupe de paramètres personnalisés Valkey en entrée doit avoir les mêmes valeurs de paramètres statiques Redis OSS que le groupe de paramètres personnalisés Redis OSS existant.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-global-replication-group \
   --global-replication-group-id myGlobalReplGroup \
   --engine valkey \
   --engine-version 8.0 \
   --apply-immediately \
   --cache-parameter-group-name myParamGroup
```

Pour Windows :

```
aws elasticache modify-global-replication-group ^
   --global-replication-group-id myGlobalReplGroup ^
   --engine valkey ^
   --engine-version 8.0 ^
   --apply-immediately ^
   --cache-parameter-group-name myParamGroup
```

- delete-global-replication-group

```
aws elasticache delete-global-replication-group \
   --global-replication-group-id my global datastore  \
   --retain-primary-replication-group defaults to true
```

- disassociate-global-replication-group

```
aws elasticache disassociate-global-replication-group \
   --global-replication-group-id my global datastore  \
   --replication-group-id my secondary cluster  \
   --replication-group-region the AWS Region in which the secondary cluster resides
```

- failover-global-replication-group

```
aws elasticache failover-replication-group \
   --global-replication-group-id my global datastore \
   --primary-region The AWS Region of the primary cluster \
   --primary-replication-group-id  The name of the global datastore, including the
 suffix.
```

- increase-node-groups-in-global-replication-group

```
aws elasticache increase-node-groups-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name \
   --node-group-count 3
```

- decrease-node-groups-in-global-replication-group

```
aws elasticache decrease-node-groups-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name \
```

```
   --node-group-count 3
```

- rebalance-shards-in-global-groupe de réplication

```
aws elasticache rebalance-shards-in-global-replication-group \
   --apply-immediately yes \
   --global-replication-group-id global-replication-group-name
```

Utilisez l'aide pour répertorier toutes les commandes disponibles ElastiCache pour Valkey ou Redis OSS.

```
aws elasticache help
```

Vous pouvez également utiliser l'aide pour décrire une commande spécifique et en savoir plus sur son utilisation :

```
aws elasticache create-global-replication-group help
```

Haute disponibilité avec les groupes de réplication

Les clusters Amazon ElastiCache Valkey et Redis OSS à nœud unique sont des entités en mémoire dotées de services de protection des données (AOF) limités. Si votre cluster échoue pour une raison quelconque, vous perdez toutes les données du cluster. Toutefois, si vous utilisez un moteur Valkey ou Redis OSS, vous pouvez regrouper 2 à 6 nœuds dans un cluster avec des répliques où 1 à 5 nœuds en lecture seule contiennent les données de réplication du nœud principal unique du groupe. read/write Dans ce scénario, si un nœud échoue pour une raison quelconque, vous ne perdez pas toutes vos données puisque celles-ci sont répliquées sur un ou plusieurs nœuds. En raison de la latence de réplication, certaines données peuvent être perdues si c'est le read/write nœud principal qui tombe en panne.

Comme le montre le graphique suivant, la structure de réplication est contenue dans une partition (appelée groupe de nœuds dans l'API/CLI) contenue dans un cluster Valkey ou Redis OSS. Les clusters Valkey ou Redis OSS (mode cluster désactivé) ont toujours une partition. Les clusters Valkey ou Redis OSS (mode cluster activé) peuvent contenir jusqu'à 500 partitions, les données du cluster étant partitionnées entre les partitions. Vous pouvez créer un cluster contenant un nombre de partitions supérieur et un nombre de réplicas inférieur, qui conduisent à un nombre total de 90 nœuds

par cluster. Cette configuration de cluster peut contenir de 90 partitions avec 0 réplica à 15 partitions avec 5 réplicas, ce qui correspond au nombre maximal de réplicas autorisé.

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster avec ElastiCache Valkey, et avec la ElastiCache version 5.0.6 ou supérieure pour Redis OSS. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Le cluster Valkey ou Redis OSS (mode cluster désactivé) possède une partition et 0 à 5 nœuds de réplication

Si le mode Multi-AZ est activé pour le cluster avec réplicas et que le nœud principal échoue, le cluster bascule vers un réplica en lecture. Etant donné que les données sont mises à jour de façon asynchrone sur les nœuds de réplica, des données peuvent être perdues du fait de la latence des mises à jour des nœuds de réplication. Pour de plus amples informations, veuillez consulter Atténuer les défaillances lors de l'exécution de Valkey ou Redis OSS.

Rubriques

- Comprendre la réplication Valkey et Redis OSS
- Réplication : mode cluster Valkey et Redis OSS désactivé ou activé
- Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS
- Implémentation de la sauvegarde et de la synchronisation

- Création d'un groupe de réplication Valkey ou Redis OSS
- Affichage des détails d'un groupe de réplication
- Recherche des points de terminaison du groupe de réplication
- Modification d'un groupe de réplication
- Suppression d'un groupe de réplication
- Modification du nombre de réplicas
- Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé)

Comprendre la réplication Valkey et Redis OSS

Redis OSS implémente la réplication de deux manières :

- Avec une seule partition contenant toutes les données du cluster dans chaque nœud : Valkey ou Redis OSS (mode cluster désactivé)
- Avec des données partitionnées sur un maximum de 500 partitions : Valkey ou Redis OSS (mode cluster activé)

Chaque partition d'un groupe de réplication possède un seul nœud read/write principal et jusqu'à 5 nœuds de réplication en lecture seule. Vous pouvez créer un cluster contenant un nombre de partitions supérieur et un nombre de réplicas inférieur, qui conduisent à un nombre total de 90 nœuds par cluster. Cette configuration de cluster peut contenir de 90 partitions avec 0 réplica à 15 partitions avec 5 réplicas, ce qui correspond au nombre maximal de réplicas autorisé.

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur Redis OSS est 5.0.6 ou supérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Rubriques

- Valkey ou Redis OSS (mode cluster désactivé)
- Valkey ou Redis OSS (mode cluster activé)

Valkey ou Redis OSS (mode cluster désactivé)

Un cluster Valkey ou Redis OSS (mode cluster désactivé) possède une seule partition, à l'intérieur de laquelle se trouve un ensemble de nœuds : un read/write nœud principal et jusqu'à cinq nœuds

répliques secondaires en lecture seule. Chaque réplica en lecture conserve une copie des données du nœud principal du cluster. Des mécanismes de réplication asynchrones sont utilisés pour maintenir les réplicas en lecture synchronisés avec le nœud principal. Les applications peuvent lire à partir de n'importe quel nœud du cluster. Les applications ne peuvent écrire que sur le nœud principal. Les réplicas en lecture améliorent le débit de lecture et préviennent la perte de données en cas de défaillance d'un nœud.

Cluster Valkey ou Redis OSS (mode cluster désactivé) avec une seule partition et des nœuds de réplication

Vous pouvez utiliser des clusters Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds de réplication pour adapter votre solution ElastiCache afin de gérer les applications gourmandes en lecture ou de prendre en charge un grand nombre de clients lisant simultanément à partir du même cluster.

Tous les nœuds d'un cluster Valkey ou Redis OSS (mode cluster désactivé) doivent résider dans la même région.

Lorsque vous ajoutez un réplica en lecture à un cluster, toutes les données du nœud principal sont copiées dans le nouveau nœud. A partir de ce moment, chaque fois que des données sont écrites sur le nœud principal, les changements sont propagés en mode asynchrone sur tous les réplicas en lecture.

Pour améliorer la tolérance aux pannes et réduire les temps d'arrêt d'écriture, activez le mode multi-AZ avec basculement automatique pour votre cluster Valkey ou Redis OSS (mode cluster désactivé) avec des répliques. Pour de plus amples informations, veuillez consulter Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS.

Vous pouvez modifier les rôles des nœuds au sein du cluster Valkey ou Redis OSS (mode cluster désactivé), le principal et l'une des répliques échangeant des rôles. Vous pouvez décider de le faire pour améliorer des performances. Par exemple, avec une application Web qui a une activité d'écriture intensive, vous pouvez choisir le nœud avec la plus basse latence réseau. Pour de plus

amples informations, veuillez consulter Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).

Valkey ou Redis OSS (mode cluster activé)

Un cluster Valkey ou Redis OSS (mode cluster activé) est composé de 1 à 500 partitions (API/CLI : groupes de nœuds). Chaque partition dispose d'un nœud primaire et jusqu'à 5 nœuds de réplica en lecture seule. La configuration peut contenir de 90 partitions avec 0 réplica à 15 partitions avec 5 réplicas, ce qui correspond au nombre maximal de réplicas autorisé.

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur est Valkey 7.2 ou supérieure, ou Redis OSS 5.0.6 ou version ultérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Chaque réplica en lecture d'une partition conserve une copie des données du nœud principal de la partition. Des mécanismes de réplication asynchrones sont utilisés pour maintenir les réplicas en lecture synchronisés avec le nœud principal. Les applications peuvent lire à partir de n'importe quel nœud du cluster. Les applications ne peuvent écrire que sur les nœuds principaux. Les réplicas en lecture améliorent l'évolutivité de la lecture et empêchent la perte de données. Les données sont partitionnées entre les partitions d'un cluster Valkey ou Redis OSS (mode cluster activé).

Les applications utilisent le point de terminaison de configuration du cluster Valkey ou Redis OSS (mode cluster activé) pour se connecter aux nœuds du cluster. Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

Cluster Valkey ou Redis OSS (mode cluster activé) avec plusieurs partitions et nœuds de réplication

Tous les nœuds d'un cluster Valkey ou Redis OSS (mode cluster activé) doivent résider dans la même région. Pour améliorer la tolérance aux pannes, vous pouvez mettre en service les réplicas principaux et en lecture dans plusieurs zones de disponibilité au sein de cette région.

Actuellement, les fonctionnalités de Valkey ou Redis OSS (mode cluster activé) présentent certaines limites.

- Vous ne pouvez promouvoir manuellement aucun des nœuds de réplica en tant que nœud principal.

Réplication : mode cluster Valkey et Redis OSS désactivé ou activé

À partir de Valkey 7.2 et de Redis OSS version 3.2, vous avez la possibilité de créer l'un des deux types distincts de clusters (API/CLI: replication groups). A Valkey or Redis OSS (cluster mode disabled) cluster always has a single shard (API/CLIgroupe de nœuds) avec jusqu'à 5 nœuds de réplication en lecture. Un cluster Valkey ou Redis OSS (mode cluster activé) comporte jusqu'à 500 partitions contenant chacune 1 à 5 nœuds de réplication en lecture.

Clusters Valkey ou Redis OSS (mode cluster désactivé) et clusters Valkey ou Redis OSS (mode cluster activé)

Le tableau suivant résume les différences importantes entre les clusters Valkey ou Redis OSS (mode cluster désactivé) et les clusters Valkey ou Redis OSS (mode cluster activé).

Comparaison de clusters Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé)

Fonctionnalité	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
Adaptabilité	Oui. Prend en charge l'ajout et la suppression des nœuds de réplica et la mise à l'échelle du type de nœud.	Limité. Pour plus d'informa tions, consultez Gestion des versions pour ElastiCache et Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).
Partitionnement de données	Non	Oui
Partitions	1	1 à 500
Réplicas en lecture	0 à 5	0 sur 5 par partition.

Fonctionnalité	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
	Important Si vous n'avez aucun réplica et que le nœud échoue, vous risquez de perdre toutes les données.	**Important** Si vous n'avez aucun réplica et qu'un nœud échoue, vous risquez de perdre toutes les données de cette partition.
Multi-AZ	Oui, avec au moins 1 réplica. Facultatif. Activé par défaut.	Oui Facultatif. Activé par défaut.
Instantanés (Sauvegardes)	Oui, création d'un fichier .rdb unique.	Oui, création d'un fichier .rdb unique pour chaque partition.
Restaurer	Oui, en utilisant un seul fichier .rdb provenant d'un cluster Valkey ou Redis OSS (mode cluster désactivé).	Oui, en utilisant des fichiers .rdb provenant d'un Valkey ou Redis OSS (mode cluster désactivé) ou d'un cluster Valkey ou Redis OSS (mode cluster activé).
Pris en charge par	Toutes les versions de Valkey et Redis OSS	Toutes les versions de Valkey, ainsi que Redis OSS 3.2 et versions ultérieures
Moteur évolutif	Oui, avec certaines limites. Pour de plus amples informations, veuillez consulter Gestion des versions pour ElastiCache.	Oui, avec certaines limites. Pour de plus amples informations, veuillez consulter Gestion des versions pour ElastiCache.

Fonctionnalité	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
Chiffrement	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.
Éligible HIPAA	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.
Conforme à PCI DSS	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.	Versions 3.2.6 (dont la fin de vie est prévue, voir le calendrier de fin de vie des versions Redis OSS) et 4.0.10 et versions ultérieures.
Repartitionnement en ligne	N/A	Version 3.2.10 (prévue pour EOL, voir le calendrier de fin de vie des versions Redis OSS) et versions ultérieures.

Lequel choisir ?

Lorsque vous choisissez entre Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé), tenez compte des facteurs suivants :

- Mise à l'échelle ou partitionnement – Les besoins métier évoluent. Vous devez soit allouer les ressources pour les fortes demandes ou mettre à l'échelle vos ressources en fonction des demandes. Valkey ou Redis OSS (mode cluster désactivé) prend en charge le dimensionnement. Vous pouvez mettre à l'échelle la capacité de lecture en ajoutant ou en supprimant des nœuds de réplica, ou vous pouvez mettre à l'échelle la capacité en optant pour un type de nœud de plus

grande capacité. Ces deux opérations prennent du temps. Pour plus d'informations, consultez Dimensionnement des nœuds de réplication pour Valkey ou Redis OSS (mode cluster désactivé).

Valkey ou Redis OSS (mode cluster activé) prend en charge le partitionnement de vos données sur un maximum de 500 groupes de nœuds. Vous pouvez modifier dynamiquement le nombre de partitions au fur et à mesure que les besoins de votre entreprise évoluent. Le partitionnement présente notamment l'avantage de pouvoir répartir votre charge sur un plus grand nombre de points de terminaison, ce qui réduit les goulots d'étranglement lorsque la demande est la plus élevée. En outre, vous pouvez gérer un plus grand nombre de données dans la mesure où elles peuvent être réparties sur plusieurs serveurs. Pour plus d'informations sur la mise à l'échelle de vos partitions, consultezMise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).

- Taille du nœud par rapport au nombre de nœuds : étant donné qu'un cluster Valkey ou Redis OSS (mode cluster désactivé) ne possède qu'une seule partition, le type de nœud doit être suffisamment grand pour accueillir toutes les données du cluster ainsi que la surcharge nécessaire. D'autre part, étant donné que vous pouvez partitionner vos données sur plusieurs partitions lorsque vous utilisez un cluster Valkey ou Redis OSS (mode cluster activé), les types de nœuds peuvent être plus petits, mais vous en avez besoin d'un plus grand nombre. Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

- Lectures et écritures : si la charge principale de votre cluster est constituée d'applications lisant des données, vous pouvez redimensionner un cluster Valkey ou Redis OSS (mode cluster désactivé) en ajoutant et en supprimant des répliques de lecture. Néanmoins, notez qu'il y a un maximum de 5 réplicas en lecture. Si la charge de votre cluster est importante en écriture, vous pouvez bénéficier des points de terminaison d'écriture supplémentaires d'un cluster Valkey ou Redis OSS (mode cluster activé) avec plusieurs partitions.

Quel que soit le type de cluster que vous choisissez d'implémenter, assurez-vous de choisir un type de nœud qui convient à vos besoins actuels et futurs.

Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS

Dans un certain nombre de cas ElastiCache , Valkey et Redis OSS peuvent 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.

Ce remplacement entraîne un certain temps d'arrêt pour le cluster, mais si Multi-AZ est activé, le temps d'arrêt est réduit. Le rôle du nœud primaire bascule automatiquement sur l'un des réplicas en lecture. Il n'est pas nécessaire de créer et de provisionner un nouveau nœud principal, car il ElastiCache 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.

ElastiCache propage également le nom DNS (Domain Name Service) de la réplique promue. De cette façon, si votre application écrit dans le point de terminaison principal, aucun changement du point de terminaison ne sera nécessaire dans le cadre de votre application. Si vous lisez à partir de points de terminaison individuels, veillez à modifier le point de terminaison de lecture du réplica promu en principal en point de terminaison du nouveau réplica.

Dans le cas de remplacements de nœuds planifiés, initiés en raison de mises à jour de maintenance ou de mises à jour en libre service, soyez conscient des points suivants :

- Pour les clusters Valkey et Redis OSS, les remplacements de nœuds prévus sont terminés pendant que le cluster traite les demandes d'écriture entrantes.
- Pour les clusters désactivés en mode cluster Valkey et Redis OSS avec le mode multi-AZ activé qui s'exécutent sur le moteur 5.0.6 ou version ultérieure, les remplacements de nœuds prévus sont terminés pendant que le cluster traite les demandes d'écriture entrantes.
- Pour les clusters désactivés en mode cluster Valkey et Redis OSS avec le mode multi-AZ activé qui s'exécutent sur le moteur 4.0.10 ou une version antérieure, vous remarquerez peut-être une brève interruption d'écriture associée aux mises à jour DNS. Cette interruption peut prendre jusqu'à quelques secondes. Ce processus est nettement plus rapide que celui qui consiste à recréer et mettre en service un nouveau nœud primaire, processus appliqué si vous n'activez pas le mode Multi-AZ.

Vous pouvez activer le Multi-AZ à l'aide de la console de ElastiCache gestion AWS CLI, de ou de l' ElastiCache API.

L'activation de ElastiCache Multi-AZ sur votre cluster Valkey ou Redis OSS (dans l'API et la CLI, groupe de réplication) améliore votre tolérance aux pannes. Cela est particulièrement vrai dans les cas où le cluster read/write principal de votre cluster devient inaccessible ou tombe en panne pour une raison quelconque. Le multi-AZ n'est pris en charge que sur les clusters Valkey et Redis OSS comportant plus d'un nœud par partition.

Rubriques

- Activation du multi-AZ
- Scénarios de défaillance avec réponses multi-AZ
- Test du basculement automatique
- Limitations relatives à la technologie Multi-AZ

Activation du multi-AZ

Vous pouvez activer le mode multi-AZ lorsque vous créez ou modifiez un cluster (API ou CLI, groupe de réplication) à l'aide de la ElastiCache console ou de l' ElastiCacheAPI. AWS CLI

Vous pouvez activer le mode multi-AZ uniquement sur les clusters Valkey ou Redis OSS (mode cluster désactivé) disposant d'au moins une réplique en lecture disponible. Les clusters sans réplica en lecture n'offrent pas une haute disponibilité ni la tolérance aux pannes. Pour plus d'informations sur la création d'un cluster avec réplication, consultez Création d'un groupe de réplication Valkey ou Redis OSS. Pour plus d'informations sur l'ajout d'un réplica à un cluster avec réplication, consultez Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé).

Rubriques

- Activation du multi-AZ (console)
- Activation de Multi-AZ (AWS CLI)
- Activation du Multi-AZ (ElastiCache API)

Activation du multi-AZ (console)

Vous pouvez activer le Multi-AZ à l'aide de la ElastiCache console lorsque vous créez un nouveau cluster Valkey ou Redis OSS ou en modifiant un cluster existant par réplication.

Le multi-AZ est activé par défaut sur les clusters Valkey ou Redis OSS (mode cluster activé).

> **Important**
>
> ElastiCache activera automatiquement le mode multi-AZ uniquement si le cluster contient au moins une réplique dans une zone de disponibilité différente de la principale dans toutes les partitions.

Activation du mode multi-AZ lors de la création d'un cluster à l'aide de la console ElastiCache

Pour plus d'informations sur ce processus, consultez Création d'un cluster Valkey (mode cluster désactivé) (console). Assurez-vous d'avoir un ou plusieurs réplicas et activez Multi-AZ.

Activation du multi-AZ sur un cluster existant (console)

Pour plus d'informations sur ce processus, consultez Modification d'un cluster À l'aide du ElastiCache AWS Management Console.

Activation de Multi-AZ (AWS CLI)

L'exemple de code suivant utilise le AWS CLI pour activer le mode multi-AZ pour le groupe `redis12` de réplication.

> **Important**
>
> Le groupe de réplication `redis12` doit déjà exister et avoir au moins un réplica en lecture disponible.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id redis12 \
    --automatic-failover-enabled \
    --multi-az-enabled \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id redis12 ^
```

```
    --automatic-failover-enabled ^
    --multi-az-enabled ^
    --apply-immediately
```

Le résultat JSON de cette commande devrait ressembler à cet exemple.

```
{
    "ReplicationGroup": {
        "Status": "modifying",
        "Description": "One shard, two nodes",
        "NodeGroups": [
            {
                "Status": "modifying",
                "NodeGroupMembers": [
                    {
                        "CurrentRole": "primary",
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis12-001"
                    },
                    {
                        "CurrentRole": "replica",
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis12-002"
                    }
                ],
                "NodeGroupId": "0001",
                "PrimaryEndpoint": {
                    "Port": 6379,
                    "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"
                }
            }
```

```
        ],
        "ReplicationGroupId": "redis12",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabling",
        "MultiAZ": "enabled",
        "SnapshotWindow": "07:00-08:00",
        "SnapshottingClusterId": "redis12-002",
        "MemberClusters": [
            "redis12-001",
            "redis12-002"
        ],
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations, consultez ces rubriques dans la Référence des commandes de l'AWS CLI :

- create-cache-cluster
- create-replication-group
- modify-replication-groupdans la référence de AWS CLI commande.

Activation du Multi-AZ (ElastiCache API)

L'exemple de code suivant utilise l' ElastiCache API pour activer le mode multi-AZ pour le groupe `redis12` de réplication.

> **Note**
>
> Pour que vous puissiez utiliser cet exemple, le groupe de réplication `redis12` doit déjà exister et avoir au moins un réplica en lecture disponible.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &AutoFailover=true
    &MultiAZEnabled=true
    &ReplicationGroupId=redis12
    &Version=2015-02-02
    &SignatureVersion=4
```

```
    &SignatureMethod=HmacSHA256
    &Timestamp=20140401T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez ces rubriques dans la référence de l'API ElastiCache :

- CreateCacheCluster
- CreateReplicationGroup
- ModifyReplicationGroup

Scénarios de défaillance avec réponses multi-AZ

Avant l'introduction du Multi-AZ, les nœuds défaillants d'un cluster étaient ElastiCache détectés et remplacés en recréant et en reprovisionnant le nœud défaillant. Si vous activez Multi-AZ, un cluster principal défaillant bascule vers le réplica ayant le moindre décalage de réplication. Le réplica sélectionné est automatiquement promu au rang de principal, ce qui est beaucoup plus rapide que de créer et de remettre en service un nouveau nœud principal. Ce processus dure généralement quelques secondes pendant lesquelles vous ne pouvez pas écrire sur le cluster.

Lorsque le mode Multi-AZ est activé, il surveille ElastiCache 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 primaire et de certains réplicas en lecture
- 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 échoue, le réplica en lecture ayant le moindre décalage de réplication est promu au rang de principal. Un réplica en lecture de remplacement est ensuite créé et provisionné dans la même zone de disponibilité que le principal défaillant.

Lorsque seul le nœud principal tombe en panne, ElastiCache Multi-AZ effectue les opérations suivantes :

1. Le nœud principal défaillant est mis hors ligne.
2. Le réplica en lecture ayant le moindre décalage de réplication est promu au rang de principal.

   Les écritures peuvent reprendre dès que le processus de promotion est terminé, généralement au bout de quelques secondes. Si votre application écrit sur le point de terminaison principal, il n'est pas nécessaire de changer le point de terminaison pour les écritures et les lectures dans la mesure où ElastiCache diffuse le nom DNS du réplica promu.
3. Un réplica en lecture de remplacement est lancé et mis en service.

   Le réplica en lecture de remplacement est lancé dans la zone de disponibilité où se trouvait le nœud principal défaillant afin que la distribution de nœuds soit maintenue.

4. La synchronisation des autres réplicas avec le nouveau nœud principal se produit.

Lorsque le nouveau réplica est disponible, vous devez être conscient des effets suivants :

- Point de terminaison principal : vous n'avez besoin d'effectuer aucune modification sur votre application, dans la mesure où le nom DNS du nouveau nœud primaire est propagé au point de terminaison principal.
- Point de terminaison de lecture : le point de terminaison du lecteur est automatiquement mis à jour de manière à pointer vers les nouveaux nœuds de réplica.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez les rubriques suivantes :

- Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)
- Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI
- Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

Scénarios de défaillance en cas de défaillance du nœud primaire et de certains réplicas en lecture

Lorsque le nœud principal et au moins un réplica en lecture sont défaillants, le réplica disponible avec le moindre décalage de réplication est promu cluster principal. De nouveaux réplicas en lecture sont également créés et mis en service dans les mêmes zones de disponibilité que les nœuds défaillants et le réplica promu cluster principal.

Lorsque le nœud principal et certaines répliques de lecture échouent, ElastiCache Multi-AZ effectue les opérations suivantes :

1. Le nœud principal et les réplicas en lecture ayant échoué sont mis hors ligne.
2. Le réplica en lecture disponible ayant le moindre décalage de réplication est promu au rang de nœud principal.

Les écritures peuvent reprendre dès que le processus de promotion est terminé, généralement au bout de quelques secondes. Si votre application écrit sur le point de terminaison principal, il n'est pas nécessaire de modifier le point de terminaison pour les écritures. ElastiCache propage le nom DNS de la réplique promue.

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 clusters sont synchronisés avec le nouveau nœud principal.

Vous devez apporter les modifications suivantes à votre application, une fois que les nouveaux nœuds sont disponibles :

- Point de terminaison principal : n'apportez aucune modification à votre application. Le nom DNS du nouveau nœud principal est propagé au point de terminaison principal.
- Point de terminaison de lecture : le point de terminaison de lecture est automatiquement mis à jour de manière à pointer vers les nouveaux nœuds de réplica.

Pour plus d'informations sur la recherche des points de terminaison d'un groupe de réplication, consultez les rubriques suivantes :

- Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)
- Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI
- Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

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.

Dans ce scénario, toutes les données du cluster sont perdues en raison de la défaillance au niveau de chaque nœud du cluster. Cela se produit rarement.

Lorsque l'ensemble du cluster échoue, ElastiCache Multi-AZ effectue les opérations suivantes :

1. Le nœud principal et les réplicas en lecture ayant échoué sont mis hors ligne.
2. Un nœud principal de remplacement est créé et mis en service.
3. Des réplicas de remplacement sont créés et provisionnés.

   Les remplacements sont créés dans les zones de disponibilité des nœuds ayant échoué afin que la distribution des nœuds soit maintenue.

   Etant donné que la totalité du cluster a échoué, les données sont perdues et tous les nouveaux nœuds démarrent à vide.

Comme chacun des nœuds de remplacement a le même point de terminaison que le nœud qu'il remplace, il n'est pas nécessaire de modifier le point de terminaison de votre application.

Pour plus d'informations sur la recherche des points de terminaison d'un groupe de réplication, consultez les rubriques suivantes :

- Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)
- Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI
- Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

Nous vous recommandons de créer le nœud principal et les réplicas dans différentes zones de disponibilité pour augmenter votre niveau de tolérance aux pannes.

Test du basculement automatique

Après avoir activé le basculement automatique, vous pouvez le tester à l'aide de la ElastiCache console AWS CLI, du et de l' ElastiCache API.

Lors du test, tenez compte des points suivants :

- Vous pouvez utiliser cette opération pour tester le basculement automatique sur un maximum de 15 partitions (appelées groupes de nœuds dans l' ElastiCache API et AWS CLI) sur une période continue de 24 heures.
- Si vous appelez cette opération sur des partitions de différents clusters (nommés groupes de réplication dans l'API et la CLI), vous pouvez effectuer des appels simultanément.
- Dans certains cas, vous pouvez effectuer cette opération plusieurs fois sur différents fragments du même groupe de réplication Valkey ou Redis OSS (mode cluster activé). 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 ElastiCache console Amazon, de l' AWS CLI API ou de l' ElastiCache API. Recherchez les événements suivants liés au basculement automatique, répertoriés ici dans leur ordre d'occurrence probable :
  1. Message du groupe de réplication : `Test Failover API called for node group <node-group-id>`
  2. Message du cluster de cache : `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  3. Message du groupe de réplication : `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  4. Message du cluster de cache : `Recovering cache nodes <node-id>`
  5. Message du cluster de cache : `Finished recovery for cache nodes <node-id>`

  Pour plus d'informations, consultez les ressources suivantes :
  - Affichage des ElastiCache événements dans le guide de l'utilisateur ElastiCache
  - DescribeEvents dans la Référence d'API ElastiCache
  - describe-events dans la Référence des commandes de l'AWS CLI .
- Cette API est conçue pour tester le comportement de votre application en cas de ElastiCache basculement. 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 ElastiCache

Test du basculement automatique à l'aide du AWS Management Console

Utilisez la procédure suivante pour tester le basculement automatique avec la console.

Pour tester le basculement automatique

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Dans la liste des clusters, cochez la case située à gauche du cluster que vous souhaitez tester. Ce cluster doit avoir au moins un nœud de réplica en lecture.
4. 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 de plus amples informations, veuillez consulter À l'aide du ElastiCache AWS Management Console.

5. Pour Valkey ou Redis OSS (mode cluster désactivé), choisissez le nom du cluster.

   Pour Valkey ou Redis OSS (mode cluster activé), procédez comme suit :

   a. Choisissez le nom du cluster.

b. Sur la page Partitions, pour la partition (nommée groupe de nœuds dans l'API ou la CLI) sur laquelle vous souhaitez tester le basculement, choisissez le nom de la partition.

6. Sur la page Nœuds, 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 (`Test Failover 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' AWS CLI aide de cette opération. `test-failover`

Paramètres

- `--replication-group-id` : obligatoire. Groupe de réplication (sur la console, cluster) qui va être testé.
- `--node-group-id` : obligatoire. Nom du groupe de nœuds sur lequel vous souhaitez tester le basculement automatique. Vous pouvez tester un maximum de 15 groupes de nœuds sur une période continue de 24 heures.

L'exemple suivant utilise le AWS CLI pour tester le basculement automatique sur le groupe `redis00-0003` de nœuds du cluster Valkey ou Redis OSS (mode cluster activé). `redis00`

Example Test du basculement automatique

Pour Linux, macOS ou Unix :

```
aws elasticache test-failover \
   --replication-group-id redis00 \
   --node-group-id redis00-0003
```

Pour Windows :

```
aws elasticache test-failover ^
   --replication-group-id redis00 ^
   --node-group-id redis00-0003
```

Le résultat de la commande précédente doit ressembler à ce qui suit.

```
{
    "ReplicationGroup": {
        "Status": "available",
        "Description": "1 shard, 3 nodes (1 + 2 replicas)",
        "NodeGroups": [
            {
                "Status": "available",
                "NodeGroupMembers": [
                    {
                        "CurrentRole": "primary",
                        "PreferredAvailabilityZone": "us-west-2c",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-001"
                    },
                    {
                        "CurrentRole": "replica",
                        "PreferredAvailabilityZone": "us-west-2a",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Port": 6379,
                            "Address":
 "redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-002"
                    },
                    {
                        "CurrentRole": "replica",
                        "PreferredAvailabilityZone": "us-west-2b",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
```

```
                            "Port": 6379,
                            "Address":
 "redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"
                        },
                        "CacheClusterId": "redis1x3-003"
                    }
                ],
                "NodeGroupId": "0001",
                "PrimaryEndpoint": {
                    "Port": 6379,
                    "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
                }
            }
        ],
        "ClusterEnabled": false,
        "ReplicationGroupId": "redis1x3",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "MultiAZ": "enabled",
        "SnapshotWindow": "11:30-12:30",
        "SnapshottingClusterId": "redis1x3-002",
        "MemberClusters": [
            "redis1x3-001",
            "redis1x3-002",
            "redis1x3-003"
        ],
        "CacheNodeType": "cache.m3.medium",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Pour suivre la progression de votre basculement, utilisez l' AWS CLI `describe-events`opération.

Pour plus d'informations, veuillez consulter les ressources suivantes :

- test-failover dans la Référence des commandes de l'AWS CLI .
- describe-events dans la Référence des commandes de l'AWS CLI .

Test du basculement automatique à l'aide de l'API ElastiCache

Vous pouvez tester le basculement automatique sur n'importe quel cluster activé avec Multi-AZ à l'aide de l'opération ElastiCache API. `TestFailover`

Paramètres

- `ReplicationGroupId` : obligatoire. Groupe de réplication (sur la console, le cluster) qui va être testé.
- `NodeGroupId` : obligatoire. Nom du groupe de nœuds sur lequel vous souhaitez tester le basculement automatique. Vous pouvez tester un maximum de 15 groupes de nœuds sur une période continue de 24 heures.

L'exemple suivant teste le basculement automatique sur le groupe de nœuds `redis00-0003` dans le groupe de réplication (sur la console, cluster) `redis00`.

Example Test du basculement automatique

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=TestFailover
    &NodeGroupId=redis00-0003
    &ReplicationGroupId=redis00
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20140401T192317Z
    &X-Amz-Credential=<credential>
```

Pour suivre la progression de votre basculement, utilisez l'opération ElastiCache `DescribeEvents` API.

Pour plus d'informations, consultez les ressources suivantes :

- TestFailoverdans la référence de ElastiCache l'API
- DescribeEventsdans la référence de ElastiCache l'API

Limitations relatives à la technologie Multi-AZ

Tenez compte des limites suivantes pour le Multi-AZ :

- Le multi-AZ est pris en charge sur Valkey et sur Redis OSS version 2.8.6 et versions ultérieures.
- Le multi-AZ n'est pas pris en charge sur les types de nœuds T1.
- La réplication Valkey et Redis OSS est asynchrone. Par conséquent, lorsqu'un nœud principal bascule vers un réplica, une petite quantité de données peut être perdue en raison d'un décalage de réplication.

  Lorsque vous choisissez le réplica à promouvoir au niveau principal, ElastiCache choisissez le réplica présentant le moins de retard de réplication. En d'autres termes, il choisit le réplica le plus à jour. Cela permet de réduire la quantité de données perdues. Le réplica dont le décalage de réplication est le moins important peut se trouver dans la même zone de disponibilité que le nœud principal défaillant ou dans une zone de disponibilité différente.
- Lorsque vous promouvez manuellement des répliques en lecture au statut principal sur des clusters Valkey ou Redis OSS avec le mode cluster désactivé, vous ne pouvez le faire que lorsque le mode multi-AZ et le basculement automatique sont désactivés. Pour promouvoir un réplica en lecture en réplica principal, procédez comme suit :
  1. Désactivez Multi-AZ sur le cluster.
  2. Désactivez le basculement automatique sur le cluster. Vous pouvez le faire via la console en décochant la case Auto failover pour le groupe de réplication. Vous pouvez également le faire en AWS CLI définissant la `AutomaticFailoverEnabled` propriété sur `false` lorsque vous appelez l'`ModifyReplicationGroup`opération.
  3. Promouvez le réplica en lecture en réplica principal.
  4. Réactivez Multi-AZ.
- ElastiCache pour Redis OSS Multi-AZ et fichier d'ajout uniquement (AOF) s'excluent mutuellement. Si vous en activez une, vous ne pouvez pas activer l'autre.
- La défaillance d'un nœud peut être provoquée par une improbable panne générale de la zone de disponibilité. Dans ce cas, le réplica remplaçant le réplica principal ayant échoué n'est créé que si la zone de disponibilité est rétablie. Par exemple, imaginons un groupe de réplication avec le principal dans AZ-a et des réplicas dans AZ-b et AZ-c. En cas de défaillance du principal, le réplica ayant le moindre décalage de réplication sera promu au rang de cluster principal. ElastiCache Crée ensuite une nouvelle réplique dans AZ-a (où se trouvait le principal défaillant) uniquement lorsque AZ-a est de nouveau disponible.

- Un redémarrage lancé par le client d'un principal n'entraîne de basculement automatique. D'autres redémarrages et défaillances déclenchent un basculement automatique.
- Lorsque le principal est redémarré, ses données sont effacées dès qu'il est à nouveau en ligne. Lorsque les réplicas en lecture voient le cluster principal effacé, ils effacent leur copie de données, ce qui entraîne une perte des données.
- Une fois qu'un réplica en lecture a été promu, les autres réplicas se synchronisent avec le nouveau principal. Après la synchronisation initiale, le contenu des réplicas est supprimé et ils synchronisent les données du nouveau principal. Ce processus de synchronisation provoque une brève interruption, au cours de laquelle les réplicas ne sont pas accessibles. Le processus de synchronisation entraîne également une augmentation de la charge temporaire sur le cluster principal lors de la synchronisation avec les réplicas. Ce comportement est propre à Valkey et Redis OSS et n'est pas propre à ElastiCache Multi-AZ. Pour plus de détails sur ce comportement, consultez la section Réplication sur le site Web de Valkey.

> **Important**
>
> Pour Valkey 7.2.6 et versions ultérieures ou Redis OSS version 2.8.22 et versions ultérieures, vous ne pouvez pas créer de répliques externes.
> Pour les versions de Redis OSS antérieures à 2.8.22, nous vous recommandons de ne pas connecter de réplique externe à un ElastiCache cluster compatible avec le mode multi-AZ. Cette configuration non prise en charge peut créer des problèmes qui ElastiCache empêchent d'effectuer correctement le basculement et la restauration. Pour connecter une réplique externe à un ElastiCache cluster, assurez-vous que le mode Multi-AZ n'est pas activé avant d'établir la connexion.

Implémentation de la sauvegarde et de la synchronisation

Toutes les versions prises en charge de Valkey et Redis OSS prennent en charge la sauvegarde et la synchronisation entre le nœud principal et le nœud de réplication. Cependant, la manière dont la sauvegarde et la synchronisation sont mises en œuvre varie en fonction de la version.

Redis OSS version 2.8.22 et versions ultérieures

La réplication Redis OSS, dans les versions 2.8.22 et ultérieures, permet de choisir entre deux méthodes. Pour plus d'informations, consultez Versions de Redis OSS antérieures à 2.8.22 et Instantané et restauration.

Au cours d'un processus sans fonction fork, si les charges d'écriture sont élevées, les écritures sur le cluster sont retardées pour éviter d'accumuler trop de changements et de mettre en échec la création de l'instantané.

Versions de Redis OSS antérieures à 2.8.22

La sauvegarde et la synchronisation de Redis OSS dans les versions antérieures à la version 2.8.22 sont un processus en trois étapes.

1. Appliquez la fonction fork et, dans le processus d'arrière-plan, sérialisez les données de cluster sur le disque. Cela crée un point-in-time instantané.
2. Au premier plan, créez un journal des modifications dans la mémoire tampon de sortie du client.

   > **Important**
   >
   > Si le journal des modifications dépasse la taille de la mémoire tampon de sortie du client, la sauvegarde ou la synchronisation échoue. Pour de plus amples informations, veuillez consulter S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS.

3. Enfin, transmettez les données de cache, puis le journal des modifications au nœud réplica.

Création d'un groupe de réplication Valkey ou Redis OSS

Vous disposez des options suivantes pour créer un cluster avec des nœuds de réplica. L'une s'applique lorsque vous disposez déjà d'un cluster Valkey ou Redis OSS (mode cluster désactivé) disponible qui n'est associé à aucun cluster contenant des répliques à utiliser comme nœud principal. L'autre s'applique lorsque vous avez besoin de créer un nœud principal avec les réplicas de cluster et en lecture. Actuellement, un cluster Valkey ou Redis OSS (mode cluster activé) doit être créé à partir de zéro.

Option 1 :Création d'un groupe de réplication à l'aide d'un cluster existant

Utilisez cette option pour tirer parti d'un cluster Valkey ou Redis OSS à nœud unique existant (mode cluster désactivé). Vous spécifiez ce nœud existant en tant que nœud primaire dans le nouveau cluster, puis ajoutez individuellement de 1 à 5 réplicas en lecture au cluster. Si le cluster existant est actif, les réplicas en lecture se synchronisent avec lui au fur et à mesure de leur création. Consultez Création d'un groupe de réplication à l'aide d'un cluster existant.

> **Important**
>
> Vous ne pouvez pas créer de cluster Valkey ou Redis OSS (mode cluster activé) en utilisant un cluster existant. Pour créer un cluster Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupe de réplication) à l'aide de la ElastiCache console, consultez. Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Option 2 :Création d'un groupe de réplication Valkey ou Redis OSS à partir de zéro

Utilisez cette option si vous ne disposez pas encore d'un cluster Valkey ou Redis OSS (mode cluster désactivé) disponible à utiliser comme nœud principal du cluster, ou si vous souhaitez créer un cluster Valkey ou Redis OSS (mode cluster activé). Consultez Création d'un groupe de réplication Valkey ou Redis OSS à partir de zéro.

Création d'un groupe de réplication à l'aide d'un cluster existant

La procédure suivante ajoute un groupe de réplication à votre cluster à nœud unique Valkey ou Redis OSS (mode cluster désactivé), ce qui est nécessaire pour mettre à niveau votre cluster vers la dernière version de Valkey. Il s'agit d'une procédure sur place qui n'implique aucun temps d'arrêt ni aucune perte de données. Lorsque vous créez un groupe de réplication pour votre cluster à nœud unique, le nœud du cluster devient le nœud principal du nouveau cluster. Si vous ne disposez pas d'un cluster Valkey ou Redis OSS (mode cluster désactivé) que vous pouvez utiliser comme cluster principal du nouveau cluster, consultez. Création d'un groupe de réplication Valkey ou Redis OSS à partir de zéro

Un cluster disponible est un cluster Valkey ou Redis OSS à nœud unique existant. Actuellement, Valkey ou Redis OSS (mode cluster activé) ne prennent pas en charge la création d'un cluster avec des répliques à l'aide d'un cluster à nœud unique disponible. Si vous souhaitez créer un cluster Valkey ou Redis OSS (mode cluster activé), consultez. Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Création d'un groupe de réplication à l'aide d'un cluster existant (console)

Consultez la rubrique En utilisant le ElastiCache AWS Management Console.

Création d'un groupe de réplication à l'aide d'un cluster de cache Valkey ou Redis OSS disponible ()AWS CLI

La création d'un groupe de réplication avec des répliques de lecture se fait en deux étapes lorsque vous utilisez un cluster de cache Valkey ou Redis OSS disponible pour le serveur principal lorsque vous utilisez le. AWS CLI

Lorsque vous utilisez le, AWS CLI vous créez un groupe de réplication en spécifiant le nœud autonome disponible comme nœud principal du cluster, `--primary-cluster-id` ainsi que le nombre de nœuds que vous souhaitez dans le cluster à l'aide de la commande CLI,`create-replication-group`. Incluez les paramètres suivants.

--replication-group-id

Le nom du groupe de réplication que vous créez. La valeur de ce paramètre sert de base pour les noms des nœuds ajoutés avec une séquence de 3 chiffres ajoutés à la fin du `--replication-group-id`. Par exemple, `sample-repl-group-001`.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) 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.

--replication-group-description

Description du groupe de réplication.

--num-node-groups

Nombre de nœuds que vous souhaitez dans ce cluster. Cette valeur inclut le nœud principal. Ce paramètre a une valeur maximale de six.

--primary-cluster-id

Nom du nœud du cluster Valkey ou Redis OSS (mode cluster désactivé) disponible dont vous souhaitez faire le nœud principal de ce groupe de réplication.

La commande suivante crée le groupe de réplication `sample-repl-group` en utilisant le cluster Valkey ou Redis OSS (mode cluster désactivé) disponible `redis01` comme nœud principal du groupe de réplication. Cela crée 2 nœuds qui sont des réplicas en lecture seule. Les paramètres de `redis01` (c'est-à-dire, le groupe de paramètres, le groupe de sécurité, le type de nœud, la version du moteur, etc.) seront appliqués à tous les nœuds du groupe de réplication.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "demo cluster with replicas" \
   --num-cache-clusters 3 \
   --primary-cluster-id redis01
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "demo cluster with replicas" ^
   --num-cache-clusters 3 ^
   --primary-cluster-id redis01
```

Pour plus d'informations et de paramètres que vous pourriez vouloir utiliser, consultez la AWS CLI rubriquecreate-replication-group.

Ajoutez ensuite des réplicas en lecture à ce groupe de réplication.

Après la création du groupe de réplication, ajoutez au groupe de un à cinq réplicas en lecture à l'aide de la commande `create-cache-cluster`, sans oublier de définir les paramètres suivants.

--cache-cluster-id

Le nom du cluster que vous ajoutez au groupe de réplication.

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.

--replication-group-id

Le nom du groupe de réplication auquel vous ajoutez ce cluster de cache.

Répétez cette commande pour chaque réplica en lecture que vous souhaitez ajouter au groupe de réplication, en modifiant seulement du paramètre `--cache-cluster-id`.

> **Note**
>
> Rappelez-vous qu'un groupe de réplication ne peut pas avoir plus de cinq réplicas en lecture. Si vous essayez d'ajouter un réplica en lecture à un groupe de réplication qui en contient déjà cinq, l'opération échoue.

Le code suivant ajoute le réplica en lecture `my-replica01` au groupe de réplication `sample-repl-group`. Les paramètres du cluster principal (groupe de paramètres, groupe de sécurité, type de nœud, etc.) seront appliqués aux nœuds à mesure qu'ils sont ajoutés au groupe de réplication.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
   --cache-cluster-id my-replica01 \
```

```
   --replication-group-id sample-repl-group
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
   --cache-cluster-id my-replica01 ^
   --replication-group-id sample-repl-group
```

Le résultat de cette commande doit être semblable à ce qui suit.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "demo cluster with replicas",
        "ClusterEnabled": false,
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "disabled",
        "SnapshotWindow": "00:00-01:00",
        "SnapshottingClusterId": "redis01",
        "MemberClusters": [
            "sample-repl-group-001",
            "sample-repl-group-002",
            "redis01"
        ],
        "CacheNodeType": "cache.m4.large",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations, consultez les AWS CLI rubriques suivantes :

- create-replication-group
- modify-replication-group

Ajouter des répliques à un cluster (API) autonome Valkey ou Redis OSS (mode cluster désactivé) ElastiCache

Lorsque vous utilisez l' ElastiCache API, vous créez un groupe de réplication en spécifiant le nœud autonome disponible en tant que nœud principal du cluster, `PrimaryClusterId` ainsi

que le nombre de nœuds que vous souhaitez inclure dans le cluster à l'aide de la commande CLI,`CreateReplicationGroup`. Incluez les paramètres suivants.

ReplicationGroupId

Le nom du groupe de réplication que vous créez. La valeur de ce paramètre sert de base pour les noms des nœuds ajoutés avec une séquence de 3 chiffres ajoutés à la fin du `ReplicationGroupId`. Par exemple, `sample-repl-group-001`.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) 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.

ReplicationGroupDescription

Description du cluster avec réplicas.

NumCacheClusters

Nombre de nœuds que vous souhaitez dans ce cluster. Cette valeur inclut le nœud principal. Ce paramètre a une valeur maximale de six.

PrimaryClusterId

Le nom du cluster Valkey ou Redis OSS disponible (mode cluster désactivé) dont vous souhaitez faire le nœud principal de ce cluster.

La commande suivante crée le cluster avec des répliques `sample-repl-group` en utilisant le cluster Valkey ou Redis OSS (mode cluster désactivé) disponible `redis01` comme nœud principal du groupe de réplication. Cela crée 2 nœuds qui sont des réplicas en lecture seule. Les paramètres de `redis01` (c'est-à-dire, le groupe de paramètres, le groupe de sécurité, le type de nœud, la version du moteur, etc.) seront appliqués à tous les nœuds du groupe de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &Engine=redis
   &EngineVersion=6.0
   &ReplicationGroupDescription=Demo%20cluster%20with%20replicas
```

```
    &ReplicationGroupId=sample-repl-group
    &PrimaryClusterId=redis01
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez les rubriques relatives aux ElastiCache APL :

- CreateReplicationGroup
- ModifyReplicationGroup

Ajoutez ensuite des réplicas en lecture à ce groupe de réplication.

Après la création du groupe de réplication, ajoutez au groupe de un à cinq réplicas en lecture en utilisant l'opération `CreateCacheCluster`, sans oublier de définir les paramètres ci-dessous.

CacheClusterId

Le nom du cluster que vous ajoutez au groupe de réplication.

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.

ReplicationGroupId

Le nom du groupe de réplication auquel vous ajoutez ce cluster de cache.

Répétez cette opération pour chaque réplica en lecture que vous souhaitez ajouter au groupe de réplication, en modifiant seulement la valeur du paramètre `CacheClusterId`.

Le code suivant ajouter le réplica en lecture `myReplica01` au groupe de réplication `myReplGroup`. Les paramètres du cluster principal (groupe de paramètres, groupe de sécurité, type de nœud, etc.) seront appliqués aux nœuds à mesure qu'ils sont ajoutés au groupe de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=CreateCacheCluster
 &CacheClusterId=myReplica01
 &ReplicationGroupId=myReplGroup
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &Version=2015-02-02
 &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
 &X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request
 &X-Amz-Date=20150202T170651Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=[signature-value]
```

Pour plus d'informations et de paramètres que vous souhaiterez peut-être utiliser, consultez la rubrique ElastiCache APICreateCacheCluster.

Création d'un groupe de réplication Valkey ou Redis OSS à partir de zéro

Vous trouverez ci-dessous comment créer un groupe de réplication Valkey ou Redis OSS sans utiliser un cluster Valkey ou Redis OSS existant comme principal. Vous pouvez créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ou Valkey ou Redis OSS (mode cluster activé) à partir de zéro à l'aide de la ElastiCache console, de l'API ou de l'API AWS CLI. ElastiCache

Avant de continuer, décidez si vous souhaitez créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ou un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Pour vous aider à faire votre choix, consultez Réplication : mode cluster Valkey et Redis OSS désactivé ou activé.

Rubriques

- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro
- Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro

Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro

Vous pouvez créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro à l'aide de la ElastiCache console, de l' AWS CLI API ou de l' ElastiCache API. Un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) possède toujours un groupe de nœuds, un cluster principal et jusqu'à cinq répliques de lecture. Les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) ne prennent pas en charge le partitionnement de vos données.

> **Note**
>
> La node/shard limite peut être augmentée jusqu'à un maximum de 500 par cluster. Pour demander une augmentation de la limite, veuillez consulter AWS Service Limits et indiquez le type d'instance dans la demande.

Pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro, adoptez l'une des approches suivantes :

Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro ()AWS CLI

La procédure suivante crée un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à l'aide du. AWS CLI

Lorsque vous créez un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro, vous créez le groupe de réplication et tous ses nœuds en un seul appel à la AWS CLI `create-replication-group` commande. Incluez les paramètres suivants.

--replication-group-id

Le nom du groupe de réplication que vous créez.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) 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.

--replication-group-description

Description du groupe de réplication.

--num-cache-clusters

Le nombre de nœuds que vous souhaitez créer avec ce groupe de réplication, réplica principal et réplicas en lecture combinés.

Si vous activez Multi-AZ (`--automatic-failover-enabled`), la valeur de `--num-cache-clusters` doit être au moins 2.

--cache-node-type

Le type de nœud pour chaque nœud dans le groupe de réplication.

ElastiCache prend en charge les types de nœuds suivants. En général, les types de la génération actuelle offrent davantage de mémoire et de puissance de calcul pour un coût inférieur, par rapport à leurs homologues équivalents de la génération précédente.

Pour plus d'informations sur les performances de chaque type de nœud, consultez Amazon EC2 Instance Types.

--data-tiering-enabled

Définissez ce paramètre si vous utilisez un type de nœud r6gd. Si vous ne souhaitez pas effectuer de hiérarchisation des données, définissez `--no-data-tiering-enabled`. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

--cache-parameter-group

Spécifiez un groupe de paramètres qui correspond à la version de votre moteur. Si vous exécutez Redis OSS 3.2.4 ou version ultérieure, spécifiez le groupe de `default.redis3.2` paramètres ou un groupe de paramètres dérivé `default.redis3.2` pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour de plus amples informations, veuillez consulter Paramètres OSS Valkey et Redis.

--network-type

`ipv4`, `ipv6` ou `dual-stack`. Si vous choisissez la double pile, vous devez définir le paramètre `--IpDiscovery` sur `ipv4` ou `ipv6`.

--engine

redis

--engine-version

Pour bénéficier de l'ensemble des fonctionnalités, choisissez la dernière version du moteur.

Les noms des nœuds dériveront du nom du groupe de réplication en ajoutant le suffixe `-00#` au nom du groupe de réplication. Par exemple, en utilisant le nom de groupe de réplication `myReplGroup`, le nom du nœud principal sera `myReplGroup-001` et pour les réplicas en lecture de `myReplGroup-002` jusqu'à `myReplGroup-006`.

Si vous souhaitez activer le chiffrement en transit ou au repos sur ce groupe de réplication, ajoutez l'un ou l'autre ou les deux paramètres `--transit-encryption-enabled` ou `--at-rest-encryption-enabled` et remplissez les conditions suivantes.

- Votre groupe de réplication doit exécuter Redis OSS version 3.2.6 ou 4.0.10.
- Le groupe de réplication doit être créé dans un Amazon VPC.
- Vous devez également inclure le paramètre `--cache-subnet-group`.
- Vous devez également inclure le paramètre `--auth-token` avec la valeur de chaîne à client spécifié pour votre jeton AUTH (mot de passe) nécessaire à la réalisation des opérations sur ce groupe de réplication.

L'opération suivante crée un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) `sample-repl-group` avec trois nœuds, un nœud principal et deux répliques.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "Demo cluster with replicas" \
   --num-cache-clusters 3 \
   --cache-node-type cache.m4.large \
   --engine redis
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "Demo cluster with replicas" ^
   --num-cache-clusters 3 ^
```

```
    --cache-node-type cache.m4.large ^
    --engine redis
```

Le résultat de la commande ressemble à ceci.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "Demo cluster with replicas",
        "ClusterEnabled": false,
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 0,
        "AutomaticFailover": "disabled",
        "SnapshotWindow": "01:30-02:30",
        "MemberClusters": [
            "sample-repl-group-001",
            "sample-repl-group-002",
            "sample-repl-group-003"
        ],
        "CacheNodeType": "cache.m4.large",
        "DataTiering": "disabled",
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations et de paramètres que vous pourriez vouloir utiliser, consultez la AWS CLI rubrique create-replication-group.

Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro (ElastiCache API)

La procédure suivante crée un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à l'aide de l' ElastiCache API.

Lorsque vous créez un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro, vous créez le groupe de réplication et tous ses nœuds en un seul appel à l'opération d' ElastiCache API`CreateReplicationGroup`. Incluez les paramètres suivants.

ReplicationGroupId

Le nom du groupe de réplication que vous créez.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster activé) 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.

ReplicationGroupDescription

La description de votre groupe de sécurité.

NumCacheClusters

Le nombre total de nœuds que vous souhaitez créer avec ce groupe de réplication, réplica principal et réplicas en lecture combinés.

Si vous activez Multi-AZ (`AutomaticFailoverEnabled=true`), la valeur de `NumCacheClusters` doit être au moins 2.

CacheNodeType

Le type de nœud pour chaque nœud dans le groupe de réplication.

ElastiCache prend en charge les types de nœuds suivants. En général, les types de la génération actuelle offrent davantage de mémoire et de puissance de calcul pour un coût inférieur, par rapport à leurs homologues équivalents de la génération précédente.

Pour plus d'informations sur les performances de chaque type de nœud, consultez Amazon EC2 Instance Types.

--data-tiering-enabled

Définissez ce paramètre si vous utilisez un type de nœud r6gd. Si vous ne souhaitez pas effectuer de hiérarchisation des données, définissez `--no-data-tiering-enabled`. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

CacheParameterGroup

Spécifiez un groupe de paramètres qui correspond à la version de votre moteur. Si vous exécutez Redis OSS 3.2.4 ou version ultérieure, spécifiez le groupe de `default.redis3.2` paramètres ou un groupe de paramètres dérivé `default.redis3.2` pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé). Pour de plus amples informations, veuillez consulter Paramètres OSS Valkey et Redis.

--network-type

`ipv4`, `ipv` ou `dual-stack`. Si vous choisissez la double pile, vous devez définir le paramètre `--IpDiscovery` sur `ipv4` ou `ipv6`.

Moteur

redis

EngineVersion

6.0

Les noms des nœuds dériveront du nom du groupe de réplication en ajoutant le suffixe `-00#` au nom du groupe de réplication. Par exemple, en utilisant le nom de groupe de réplication `myReplGroup`, le nom du nœud principal sera `myReplGroup-001` et pour les réplicas en lecture de `myReplGroup-002` jusqu'à `myReplGroup-006`.

Si vous souhaitez activer le chiffrement en transit ou au repos sur ce groupe de réplication, ajoutez l'un ou l'autre ou les deux paramètres `TransitEncryptionEnabled=true` ou `AtRestEncryptionEnabled=true` et remplissez les conditions suivantes.

- Votre groupe de réplication doit exécuter Redis OSS version 3.2.6 ou 4.0.10.
- Le groupe de réplication doit être créé dans un Amazon VPC.
- Vous devez également inclure le paramètre `CacheSubnetGroup`.
- Vous devez également inclure le paramètre `AuthToken` avec la valeur de chaîne à client spécifié pour votre jeton AUTH (mot de passe) nécessaire à la réalisation des opérations sur ce groupe de réplication.

L'opération suivante crée le groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) `myReplGroup` avec trois nœuds, un nœud principal et deux répliques.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &CacheNodeType=cache.m4.large
   &CacheParameterGroup=default.redis6.x
   &Engine=redis
   &EngineVersion=6.0
   &NumCacheClusters=3
   &ReplicationGroupDescription=test%20group
```

```
    &ReplicationGroupId=myReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations et de paramètres que vous souhaiterez peut-être utiliser, consultez la rubrique ElastiCache APICreateReplicationGroup.

Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro

Vous pouvez créer un cluster Valkey ou Redis OSS (mode cluster activé) (API/CLI : groupe de réplication) à l'aide de la ElastiCache console, de l'API ou de l' AWS CLI API. ElastiCache Un groupe de réplication Valkey ou Redis OSS (mode cluster activé) comprend de 1 à 500 partitions (API/CLI : groupes de nœuds), un nœud principal dans chaque partition et jusqu'à 5 répliques de lecture dans chaque partition. Vous pouvez créer un cluster contenant un nombre de partitions supérieur et un nombre de réplicas inférieur, qui conduisent à un nombre total de 90 nœuds par cluster. Cette configuration de cluster peut contenir de 90 partitions avec 0 réplica à 15 partitions avec 5 réplicas, ce qui correspond au nombre maximal de réplicas autorisé.

La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 500 par cluster si la version du moteur Valkey ou Redis OSS est 5.0.6 ou supérieure. 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. Pour de plus amples informations, veuillez consulter Création d'un groupe de sous-réseaux.

Pour les versions antérieures à 5.0.6, la limite est de 250 par cluster.

Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limits et sélectionnez le type de limite Nœuds par cluster par type d'instance.

Création d'un cluster dans Valkey ou Redis OSS (mode cluster activé)

- Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)
- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro ()AWS CLI
- Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro (ElastiCache API)

Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Pour créer un cluster Valkey ou Redis OSS (mode cluster activé), consultez. Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console) Veillez à activer le mode de cluster, Cluster Mode enabled (Scale Out) et spécifiez au moins deux partitions et un nœud de réplica dans chaque.

Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro ()AWS CLI

La procédure suivante crée un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à l'aide du. AWS CLI

Lorsque vous créez un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro, vous créez le groupe de réplication et tous ses nœuds en un seul appel à la AWS CLI `create-replication-group` commande. Incluez les paramètres suivants.

--replication-group-id

Le nom du groupe de réplication que vous créez.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster activé) 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.

--replication-group-description

Description du groupe de réplication.

--cache-node-type

Le type de nœud pour chaque nœud dans le groupe de réplication.

ElastiCache prend en charge les types de nœuds suivants. En général, les types de la génération actuelle offrent davantage de mémoire et de puissance de calcul pour un coût inférieur, par rapport à leurs homologues équivalents de la génération précédente.

Pour plus d'informations sur les performances de chaque type de nœud, consultez Amazon EC2 Instance Types.

--data-tiering-enabled

Définissez ce paramètre si vous utilisez un type de nœud r6gd. Si vous ne souhaitez pas effectuer de hiérarchisation des données, définissez `--no-data-tiering-enabled`. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

--cache-parameter-group

Spécifiez le groupe de `default.redis6.x.cluster.on` paramètres ou un groupe de paramètres dérivé `default.redis6.x.cluster.on` pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Pour de plus amples informations, veuillez consulter Modifications des paramètres de Redis OSS 6.x.

--engine

redis

--engine-version

3.2.4

--num-node-groups

Le nombre de groupes de nœuds dans ce groupe de réplication. Valeurs valides : de 1 à 500.

> **Note**
>
> La node/shard limite peut être augmentée jusqu'à un maximum de 500 par cluster. Pour demander une augmentation de cette limite, veuillez consulter AWS Service Limit et choisissez le type de limite « Nœuds par cluster par type d'instance ».

--replicas-per-node-group

Le nombre de nœuds de réplica dans chaque groupe de nœuds. Les valeurs valides sont comprises entre 0 et 5.

--network-type

`ipv4`, `ipv` ou `dual-stack`. Si vous choisissez la double pile, vous devez définir le paramètre `--IpDiscovery` sur `ipv4` ou `ipv6`.

Si vous souhaitez activer le chiffrement en transit ou au repos sur ce groupe de réplication, ajoutez l'un ou l'autre ou les deux paramètres `--transit-encryption-enabled` ou `--at-rest-encryption-enabled` et remplissez les conditions suivantes.

- Votre groupe de réplication doit exécuter Redis OSS version 3.2.6 ou 4.0.10.
- Le groupe de réplication doit être créé dans un Amazon VPC.

- Vous devez également inclure le paramètre `--cache-subnet-group`.
- Vous devez également inclure le paramètre `--auth-token` avec la valeur de chaîne à client spécifié pour votre jeton AUTH (mot de passe) nécessaire à la réalisation des opérations sur ce groupe de réplication.

L'opération suivante crée le groupe de réplication Valkey ou Redis OSS (mode cluster activé) `sample-repl-group` avec trois nœuds groups/shards (--num-node-groups), chacun avec trois nœuds, un principal et deux répliques en lecture (--). replicas-per-node-group

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id sample-repl-group \
   --replication-group-description "Demo cluster with replicas" \
   --num-node-groups 3 \
   --replicas-per-node-group 2 \
   --cache-node-type cache.m4.large \
   --engine redis \
   --security-group-ids SECURITY_GROUP_ID \
   --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id sample-repl-group ^
   --replication-group-description "Demo cluster with replicas" ^
   --num-node-groups 3 ^
   --replicas-per-node-group 2 ^
   --cache-node-type cache.m4.large ^
   --engine redis ^
   --security-group-ids SECURITY_GROUP_ID ^
   --cache-subnet-group-name SUBNET_GROUP_NAME>
```

La commande précédente génère la sortie suivante.

```
{
    "ReplicationGroup": {
        "Status": "creating",
```

```
        "Description": "Demo cluster with replicas",
        "ReplicationGroupId": "sample-repl-group",
        "SnapshotRetentionLimit": 0,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "05:30-06:30",
        "MemberClusters": [
            "sample-repl-group-0001-001",
            "sample-repl-group-0001-002",
            "sample-repl-group-0001-003",
            "sample-repl-group-0002-001",
            "sample-repl-group-0002-002",
            "sample-repl-group-0002-003",
            "sample-repl-group-0003-001",
            "sample-repl-group-0003-002",
            "sample-repl-group-0003-003"
        ],
        "PendingModifiedValues": {}
    }
}
```

Lorsque vous créez un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro, vous pouvez configurer chaque partition du cluster à l'aide du `--node-group-configuration` paramètre illustré dans l'exemple suivant qui configure deux groupes de nœuds (console : partitions). La première partition comporte deux nœuds, un nœud principal et un réplica en lecture. La deuxième partition comporte trois nœuds, un nœud principal et deux réplicas en lecture.

--node-group-configuration

La configuration pour chaque groupe de nœuds. Le paramètre `--node-group-configuration` se compose des champs suivants.

- `PrimaryAvailabilityZone` : la zone de disponibilité où se trouve le nœud primaire de ce groupe de nœuds. Si ce paramètre est omis, ElastiCache choisit la zone de disponibilité pour le nœud principal.

  Par exemple : us-west-2a.
- `ReplicaAvailabilityZones` : une liste séparée par des virgules des zones de disponibilité où se trouvent les réplicas en lecture. Le nombre de zones de disponibilité dans cette liste doit correspondre à la valeur `ReplicaCount`. Si ce paramètre est omis, ElastiCache choisit les zones de disponibilité pour les nœuds de réplication.

  Exemple : « us-west-2a, us-west-2b, us-west-2c »

- `ReplicaCount` – Le nombre de nœuds de réplica dans ce groupe de nœuds.
- `Slots` : chaîne qui spécifie l'espace de clés pour le groupe de nœuds. La chaîne est au format `startKey-endKey`. Si ce paramètre est omis, les clés sont réparties ElastiCache de manière égale entre les groupes de nœuds.

  Exemple : "0-4999"

L'opération suivante crée le groupe de réplication Valkey ou Redis OSS (mode cluster activé) `new-group` avec deux groupes/fragments de nœuds (). `--num-node-groups` À l'inverse de l'exemple précédent, chaque groupe de nœuds est configuré différemment de l'autre groupe de nœuds (`--node-group-configuration`).

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
  --replication-group-id new-group \
  --replication-group-description "Sharded replication group" \
  --engine redis \
  --snapshot-retention-limit 8 \
  --cache-node-type cache.m4.medium \
  --num-node-groups 2 \
  --node-group-configuration \
      "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-
east-1c',ReplicaAvailabilityZones='us-east-1b'" \
      "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-
east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

Pour Windows :

```
aws elasticache create-replication-group ^
  --replication-group-id new-group ^
  --replication-group-description "Sharded replication group" ^
  --engine redis ^
  --snapshot-retention-limit 8 ^
  --cache-node-type cache.m4.medium ^
  --num-node-groups 2 ^
  --node-group-configuration \
      "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-
east-1c',ReplicaAvailabilityZones='us-east-1b'" \
```

```
      "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-
east-1a',ReplicaAvailabilityZones='us-east-1a','us-east-1c'"
```

L'opération précédente génère la sortie suivante.

```
{
    "ReplicationGroup": {
        "Status": "creating",
        "Description": "Sharded replication group",
        "ReplicationGroupId": "rc-rg",
        "SnapshotRetentionLimit": 8,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "10:00-11:00",
        "MemberClusters": [
            "rc-rg-0001-001",
            "rc-rg-0001-002",
            "rc-rg-0002-001",
            "rc-rg-0002-002",
            "rc-rg-0002-003"
        ],
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations et de paramètres que vous pourriez vouloir utiliser, consultez la AWS CLI rubriquecreate-replication-group.

Création d'un groupe de réplication dans Valkey ou Redis OSS (mode cluster activé) à partir de zéro (ElastiCache API)

La procédure suivante crée un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à l'aide de l' ElastiCache API.

Lorsque vous créez un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro, vous créez le groupe de réplication et tous ses nœuds en un seul appel à l'opération d' ElastiCache API`CreateReplicationGroup`. Incluez les paramètres suivants.

ReplicationGroupId

Le nom du groupe de réplication que vous créez.

Les contraintes de dénomination des groupes de réplication Valkey ou Redis OSS (mode cluster activé) 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.

ReplicationGroupDescription

Description du groupe de réplication.

NumNodeGroups

Le nombre de groupes de nœuds que vous souhaitez créer avec ce groupe de réplication. Valeurs valides : de 1 à 500.

ReplicasPerNodeGroup

Le nombre de nœuds de réplica dans chaque groupe de nœuds. Les valeurs valides sont comprises entre 1 et 5.

NodeGroupConfiguration

La configuration pour chaque groupe de nœuds. Le paramètre `NodeGroupConfiguration` se compose des champs suivants.

- `PrimaryAvailabilityZone` : la zone de disponibilité où se trouve le nœud primaire de ce groupe de nœuds. Si ce paramètre est omis, ElastiCache choisit la zone de disponibilité pour le nœud principal.

  Par exemple : us-west-2a.
- `ReplicaAvailabilityZones` : une liste des zones de disponibilité dans lesquelles se trouvent les réplicas en lecture. Le nombre de zones de disponibilité dans cette liste doit correspondre à la valeur `ReplicaCount`. Si ce paramètre est omis, ElastiCache choisit les zones de disponibilité pour les nœuds de réplication.
- `ReplicaCount` : le nombre de nœuds de réplica dans ce groupe de nœuds.
- `Slots` : chaîne qui spécifie l'espace de clés pour le groupe de nœuds. La chaîne est au format `startKey-endKey`. Si ce paramètre est omis, les clés sont réparties ElastiCache de manière égale entre les groupes de nœuds.

  Exemple : "0-4999"

CacheNodeType

Le type de nœud pour chaque nœud dans le groupe de réplication.

ElastiCache prend en charge les types de nœuds suivants. En général, les types de la génération actuelle offrent davantage de mémoire et de puissance de calcul pour un coût inférieur, par rapport à leurs homologues équivalents de la génération précédente.

Pour plus d'informations sur les performances de chaque type de nœud, consultez Amazon EC2 Instance Types.

--data-tiering-enabled

Définissez ce paramètre si vous utilisez un type de nœud r6gd. Si vous ne souhaitez pas effectuer de hiérarchisation des données, définissez `--no-data-tiering-enabled`. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

CacheParameterGroup

Spécifiez le groupe de `default.redis6.x.cluster.on` paramètres ou un groupe de paramètres dérivé `default.redis6.x.cluster.on` pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Pour de plus amples informations, veuillez consulter Modifications des paramètres de Redis OSS 6.x.

--network-type

`ipv4`, `ipv` ou `dual-stack`. Si vous choisissez la double pile, vous devez définir le paramètre `--IpDiscovery` sur `ipv4` ou `ipv6`.

Moteur

redis

EngineVersion

6.0

Si vous souhaitez activer le chiffrement en transit ou au repos sur ce groupe de réplication, ajoutez l'un ou l'autre ou les deux paramètres `TransitEncryptionEnabled=true` ou `AtRestEncryptionEnabled=true` et remplissez les conditions suivantes.

- Votre groupe de réplication doit exécuter Redis OSS version 3.2.6 ou 4.0.10.
- Le groupe de réplication doit être créé dans un Amazon VPC.

- Vous devez également inclure le paramètre `CacheSubnetGroup`.
- Vous devez également inclure le paramètre `AuthToken` avec la valeur de chaîne à client spécifié pour votre jeton AUTH (mot de passe) nécessaire à la réalisation des opérations sur ce groupe de réplication.

Des sauts de ligne sont ajoutés pour faciliter la lecture.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateReplicationGroup
   &CacheNodeType=cache.m4.large
   &CacheParemeterGroup=default.redis6.xcluster.on
   &Engine=redis
   &EngineVersion=6.0
   &NumNodeGroups=3
   &ReplicasPerNodeGroup=2
   &ReplicationGroupDescription=test%20group
   &ReplicationGroupId=myReplGroup
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Pour plus d'informations et de paramètres que vous souhaiterez peut-être utiliser, consultez la rubrique ElastiCache APICreateReplicationGroup.

Affichage des détails d'un groupe de réplication

Selon vos besoins, vous pouvez afficher les détails d'un groupe de réplication. Vous pouvez utiliser la ElastiCache console, le AWS CLI for ElastiCache ou l' ElastiCache API. Le processus de console est différent pour Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé).

Affichage des détails d'un groupe de réplication

- Affichage d'un système d'exploitation Valkey ou Redis (mode cluster désactivé) avec des répliques
  - Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) (console)
  - Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ()AWS CLI
  - Affichage d'un groupe de réplication (API) Valkey ou Redis OSS (ElastiCache mode cluster désactivé)

- Affichage d'un groupe de réplication : Valkey ou Redis OSS (mode cluster activé)
  - Affichage d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)
  - Affichage d'un cluster Valkey ou Redis OSS (mode cluster activé) ()AWS CLI
  - Affichage d'un cluster (API) Valkey ou Redis OSS (mode cluster activé) ElastiCache
- Affichage des détails d'un groupe de réplication (AWS CLI)
- Afficher les détails d'un groupe de réplication (ElastiCache API)

Affichage d'un système d'exploitation Valkey ou Redis (mode cluster désactivé) avec des répliques

Vous pouvez afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des répliques (API/CLI : groupe de réplication) à l'aide de la ElastiCache console, du AWS CLI for ElastiCache ou de l'API. ElastiCache

Afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé)

- Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) (console)
- Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ()AWS CLI
- Affichage d'un groupe de réplication (API) Valkey ou Redis OSS (ElastiCache mode cluster désactivé)

Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) (console)

Pour afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des répliques à l'aide de la ElastiCache console, consultez la rubrique. Affichage des détails de Valkey ou Redis OSS (mode cluster désactivé) (console)

Affichage d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ()AWS CLI

Pour un AWS CLI exemple qui affiche les détails d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé), consultez. Affichage des détails d'un groupe de réplication (AWS CLI)

Affichage d'un groupe de réplication (API) Valkey ou Redis OSS (ElastiCache mode cluster désactivé)

Pour un exemple d' ElastiCache API qui affiche les détails d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé), consultez. Afficher les détails d'un groupe de réplication (ElastiCache API)

Affichage d'un groupe de réplication : Valkey ou Redis OSS (mode cluster activé)

Affichage d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Pour afficher les détails d'un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de la ElastiCache console, consultez. Affichage des détails d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Affichage d'un cluster Valkey ou Redis OSS (mode cluster activé) ()AWS CLI

Pour un exemple de ElastiCache CLI qui affiche les détails d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé), consultez. Affichage des détails d'un groupe de réplication (AWS CLI)

Affichage d'un cluster (API) Valkey ou Redis OSS (mode cluster activé) ElastiCache

Pour un exemple d' ElastiCache API qui affiche les détails d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé), consultez. Afficher les détails d'un groupe de réplication (ElastiCache API)

Affichage des détails d'un groupe de réplication (AWS CLI)

Vous pouvez consulter les détails d'un groupe de réplication à l'aide de la AWS CLI `describe-replication-groups` commande. Utilisez les paramètres facultatifs suivants pour affiner la liste. Si les paramètres ne sont pas spécifiés, vous obtenez les détails de 100 groupes de réplication maximum.

Paramètres facultatifs

- `--replication-group-id` – Utilisez ce paramètre pour afficher les détails d'un groupe de réplication spécifique. Si le groupe de réplication spécifié comprend plus d'un groupe de nœuds, les résultats obtenus sont regroupés par groupe de nœuds.
- `--max-items` – Utilisez ce paramètre pour limiter le nombre de groupes de réplication répertoriés. La valeur de `--max-items` doit être comprise entre 20 et 100.

Example

Le code suivant répertorie les détails de 100 groupes de réplication maximum.

```
aws elasticache describe-replication-groups
```

Le code suivant répertorie les détails de `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Le code suivant répertorie les détails de `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Le code suivant répertorie les détails de 25 groupes de réplication maximum.

```
aws elasticache describe-replication-groups --max-items 25
```

Le résultat de cette opération doit ressembler à cet exemple (format JSON).

```
{
   "ReplicationGroups": [
     {
       "Status": "available",
       "Description": "test",
       "NodeGroups": [
         {
            "Status": "available",
               "NodeGroupMembers": [
                  {
                     "CurrentRole": "primary",
                     "PreferredAvailabilityZone": "us-west-2a",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
                     },
                     "CacheClusterId": "rg-name-001"
                  },
                  {
                     "CurrentRole": "replica",
                     "PreferredAvailabilityZone": "us-west-2b",
                     "CacheNodeId": "0001",
                     "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
                     },
```

```
                    "CacheClusterId": "rg-name-002"
                },
                {
                    "CurrentRole": "replica",
                    "PreferredAvailabilityZone": "us-west-2c",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Port": 6379,
                        "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
                    },
                    "CacheClusterId": "rg-name-003"
                }
            ],
            "NodeGroupId": "0001",
            "PrimaryEndpoint": {
                "Port": 6379,
                "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
            }
        }
    ],
    "ReplicationGroupId": "rg-name",
    "AutomaticFailover": "enabled",
    "SnapshottingClusterId": "rg-name-002",
    "MemberClusters": [
        "rg-name-001",
        "rg-name-002",
        "rg-name-003"
    ],
    "PendingModifiedValues": {}
  },
  {
  ... some output omitted for brevity
  }
]
}
```

Pour plus d'informations, consultez AWS CLI la ElastiCache rubrique dédiée[describe-replication-groups](#).

Afficher les détails d'un groupe de réplication (ElastiCache API)

Vous pouvez consulter les détails d'une réplication à l'aide de l' AWS CLI `DescribeReplicationGroups`opération. Utilisez les paramètres facultatifs suivants pour affiner la

liste. Si les paramètres ne sont pas spécifiés, vous obtenez les détails de 100 groupes de réplication maximum.

Paramètres facultatifs

- `ReplicationGroupId` – Utilisez ce paramètre pour afficher les détails d'un groupe de réplication spécifique. Si le groupe de réplication spécifié comprend plus d'un groupe de nœuds, les résultats obtenus sont regroupés par groupe de nœuds.
- `MaxRecords` – Utilisez ce paramètre pour limiter le nombre de groupes de réplication répertoriés. La valeur de `MaxRecords` doit être comprise entre 20 et 100. La valeur par défaut est 100.

Example

Le code suivant répertorie les détails de 100 groupes de réplication maximum.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroups
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Le code suivant répertorie les détails de `myReplGroup`.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroups
   &ReplicationGroupId=myReplGroup
   &Version=2015-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Le code suivant affiche les détails de 25 clusters maximum.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeReplicationGroups
   &MaxRecords=25
   &Version=2015-02-02
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez la rubrique de référence sur les ElastiCache APIDescribeReplicationGroups.

Recherche des points de terminaison du groupe de réplication

Une application peut se connecter à n'importe quel nœud du groupe de réplication, à condition qu'elle dispose du point de terminaison DNS et du numéro de port de ce nœud. Selon que vous utilisez un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) ou un groupe de réplication Valkey ou Redis OSS (mode cluster activé), différents points de terminaison vous intéresseront.

Valkey ou Redis OSS (mode cluster désactivé)

Les clusters Valkey ou Redis OSS (mode cluster désactivé) dotés de répliques ont trois types de points de terminaison : le point de terminaison principal, le point de terminaison du lecteur et les points de terminaison du nœud. Le point de terminaison principal est un nom DNS qui est toujours résolu dans le nœud principal du cluster. Le point de terminaison principal n'est pas affecté par les changements apportés à votre cluster, tels que la promotion d'un réplica en lecture au rang de réplica principal. Pour l'activité d'écriture, il est préférable que vos applications se connectent au point de terminaison principal.

Un point de terminaison de lecteur répartira de manière égale les connexions entrantes vers le point de terminaison entre toutes les répliques de lecture d'un ElastiCache cluster. Des facteurs supplémentaires, comme le moment où l'application crée les connexions ou la façon dont l'application utilise ou réutilise les connexions, détermineront la distribution du trafic. Les points de terminaison du lecteur suivent les changements de cluster en temps réel au fur et à mesure que des réplicas sont ajoutés ou supprimés. Vous pouvez placer ElastiCache les multiples répliques de lecture de votre cluster Redis OSS dans différentes zones de AWS disponibilité (AZ) afin de garantir la haute disponibilité des points de terminaison des lecteurs.

> **Note**
>
> Un point de terminaison du lecteur n'est pas un équilibreur de charge. Il s'agit d'un enregistrement DNS qui se résoudra à une adresse IP de l'un des nœuds de réplica en mode round robin.

Pour l'activité de lecture, les applications peuvent également se connecter à n'importe quel nœud du cluster. Contrairement au point de terminaison principal, les points de terminaison des nœuds sont résolus dans des points de terminaison spécifiques. En cas de modification dans votre cluster, comme l'ajout ou la suppression d'un réplica, vous devez mettre à jour les points de terminaison des nœuds de votre application.

Valkey ou Redis OSS (mode cluster activé)

Les clusters Valkey ou Redis OSS (mode cluster activé) avec des répliques, parce qu'ils comportent plusieurs partitions (API/CLI : groupes de nœuds), ce qui signifie qu'ils possèdent également plusieurs nœuds principaux, ont une structure de point de terminaison différente de celle des clusters Valkey ou Redis OSS (mode cluster désactivé). Valkey ou Redis OSS (mode cluster activé) possède un point de terminaison de configuration qui « connaît » tous les points de terminaison principaux et nœuds du cluster. Votre application se connecte au point de terminaison de configuration. Chaque fois que votre application écrit ou lit depuis le point de terminaison de configuration du cluster, Valkey et Redis OSS déterminent en arrière-plan à quelle partition appartient la clé et quel point de terminaison de cette partition utiliser. Ce processus est transparent pour votre application.

Vous pouvez trouver les points de terminaison d'un cluster à l'aide de la ElastiCache console AWS CLI, de ou de l' ElastiCache API.

Recherche des points de terminaison du groupe de réplication

Pour trouver les points de terminaison de votre groupe de réplication, consultez l'une des rubriques suivantes :

- Trouver les points de terminaison d'un cluster Valkey ou Redis OSS (mode cluster désactivé) (console)
- Recherche de points de terminaison pour un cluster Valkey ou Redis OSS (mode cluster activé) (console)
- Recherche des points de terminaison pour les groupes de réplication Valkey ou Redis OSS ()AWS CLI
- Recherche de points de terminaison pour les groupes de réplication (API) Valkey ou Redis OSS ElastiCache

Modification d'un groupe de réplication

⚠ Contraintes importantes

- Actuellement, ElastiCache prend en charge des modifications limitées d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé), par exemple la modification de la version du moteur à l'aide de l'opération API (`ModifyReplicationGroup`CLI :`modify-replication-group`). Vous pouvez modifier le nombre de partitions (groupes de nœuds) dans un cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de l'opération API (`ModifyReplicationGroupShardConfiguration`CLI :). `modify-replication-group-shard-configuration` Pour de plus amples informations, veuillez consulter Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé).

  Les autres modifications apportées à un cluster Valkey ou Redis OSS (mode cluster activé) nécessitent que vous créiez un cluster avec le nouveau cluster incorporant les modifications.
- Vous pouvez mettre à niveau les clusters et groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) et Valkey ou Redis OSS (mode cluster activé) vers des versions de moteur plus récentes. Cependant, vous ne pouvez pas mettre à niveau vers des versions antérieures du moteur, sauf en supprimant le cluster existant ou le groupe de réplication et en le créant à nouveau. Pour de plus amples informations, veuillez consulter Gestion des versions pour ElastiCache.
- Vous pouvez mettre à niveau un cluster existant ElastiCache pour Valkey ou Redis OSS qui utilise le mode cluster désactivé pour utiliser le mode cluster activé, à l'aide de la console, de l'ModifyReplicationGroupAPI ou de la commande modify-replication-groupCLI, comme indiqué dans l'exemple ci-dessous. Vous pouvez également suivre les étapes décrites dans Modifying cluster mode (Modification du mode cluster).

Vous pouvez modifier les paramètres d'un cluster Valkey ou Redis OSS (mode cluster désactivé) à l'aide de la ElastiCache console, de l' AWS CLI API ou de l' ElastiCache API. Actuellement, ElastiCache prend en charge un nombre limité de modifications sur un groupe de réplication Valkey ou Redis OSS (mode cluster activé). D'autres modifications nécessitent que vous créiez une sauvegarde du groupe de réplication actuel, puis que vous utilisiez cette sauvegarde pour démarrer un nouveau groupe de réplication Valkey ou Redis OSS (mode cluster activé).

Rubriques

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

À l'aide du AWS Management Console

Pour modifier un cluster Valkey ou Redis OSS (mode cluster désactivé), consultez. Modification d'un ElastiCache cluster

À l'aide du AWS CLI

Voici des AWS CLI exemples de `modify-replication-group` commande. Vous pouvez utiliser la même commande pour apporter d'autres modifications à un groupe de réplication.

Activez le Multi-AZ sur un groupe de réplication Valkey ou Redis OSS existant :

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --multi-az-enabled = true
```

Pour Windows :

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --multi-az-enabled
```

Modifier le mode cluster de désactivé à activé :

Pour modifier le mode cluster de Désactivé à Activé, vous devez d'abord définir le mode cluster comme Compatible. Le mode compatible permet à vos clients Valkey ou Redis OSS de se connecter en utilisant à la fois le mode cluster activé et le mode cluster désactivé. Après avoir migré tous les clients Valkey ou Redis OSS pour utiliser le mode cluster activé, vous pouvez terminer la configuration du mode cluster et définir le mode cluster sur activé.

Pour Linux, macOS ou Unix :

Définissez le mode cluster sur Compatible.

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --cache-parameter-group-name myParameterGroupName \
   --cluster-mode compatible
```

Définissez le mode cluster sur Activé.

```
aws elasticache modify-replication-group \
   --replication-group-id myReplGroup \
   --cluster-mode enabled
```

Pour Windows :

Définissez le mode cluster sur Compatible.

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --cache-parameter-group-name myParameterGroupName ^
   --cluster-mode compatible
```

Définissez le mode cluster sur Activé.

```
aws elasticache modify-replication-group ^
   --replication-group-id myReplGroup ^
   --cluster-mode enabled
```

Pour plus d'informations sur la AWS CLI `modify-replication-group` commande, consultez modify-replication-group la section Modification du mode cluster dans le Guide de l'ElastiCache utilisateur de Redis OSS.

Utilisation de l' ElastiCache API

L'opération ElastiCache d'API suivante active le multi-AZ sur un groupe de réplication Valkey ou Redis OSS existant. Vous pouvez utiliser la même opération pour apporter d'autres modifications à un groupe de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyReplicationGroup
   &AutomaticFailoverEnabled=true
   &Mutli-AZEnabled=true
```

```
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations sur le `ModifyReplicationGroup` fonctionnement de l' ElastiCache API, consultezModifyReplicationGroup.

Suppression d'un groupe de réplication

Si vous n'avez plus besoin d'un de vos clusters avec réplicas (appelés groupes de réplication dans l'API/CLI), vous pouvez le supprimer. Lorsque vous supprimez un groupe de réplication, tous ElastiCache les nœuds de ce groupe sont supprimés.

Une fois que vous avez commencé cette opération, elle ne peut être ni interrompue ni annulée.

> **Warning**
>
> - Lorsque vous supprimez un ElastiCache cluster Redis OSS, vos instantanés manuels sont conservés. Vous avez également la possibilité de créer un instantané final avant la suppression du cluster. Les instantanés mis en cache automatiquement ne sont pas conservés.
> - `CreateSnapshot`une autorisation est requise pour créer un instantané final. Sans cette autorisation, l'appel d'API échouera avec une `Access Denied` exception.

Suppression d'un groupe de réplication (console)

Pour supprimer un cluster qui contient des réplicas, consultez Supprimer un cluster dans ElastiCache.

Suppression d'un groupe de réplication (AWS CLI)

Utilisez la commande delete-replication-group pour supprimer un groupe de réplication.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Une invite vous demande de confirmer votre décision. Entrez y (oui) pour démarrer l'opération immédiatement. Une fois le processus entamé, il est irréversible.

```
  After you begin deleting this replication group, all of its nodes will be deleted as
 well.
  Are you sure you want to delete this replication group? [Ny]y

REPLICATIONGROUP  my-repgroup  My replication group  deleting
```

Supprimer un groupe de réplication (ElastiCache API)

Appelez DeleteReplicationGroup avec le paramètre `ReplicationGroup`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteReplicationGroup
   &ReplicationGroupId=my-repgroup
   &Version=2014-12-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20141201T220302Z
   &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
   &X-Amz-Date=20141201T220302Z
   &X-Amz-SignedHeaders=Host
   &X-Amz-Expires=20141201T220302Z
   &X-Amz-Credential=<credential>
   &X-Amz-Signature=<signature>
```

> **Note**
>
> Si vous définissez le paramètre `RetainPrimaryCluster` sur `true`, tous les réplicas en lecture seront supprimés, mais le cluster principal sera conservé.

Modification du nombre de réplicas

Vous pouvez augmenter ou diminuer dynamiquement le nombre de répliques de lecture dans votre groupe de réplication Valkey ou Redis OSS à l'aide de l'API AWS Management Console, de ou de l' AWS CLI API. ElastiCache Si votre groupe de réplication est un groupe de réplication Valkey ou Redis OSS (mode cluster activé), vous pouvez choisir les partitions (groupes de nœuds) pour augmenter ou diminuer le nombre de répliques.

Pour modifier dynamiquement le nombre de répliques dans votre groupe de réplication, choisissez l'opération adaptée à votre situation dans le tableau suivant.

Pour ce faire	Pour Valkey ou Redis OSS (mode cluster activé)	Pour Valkey ou Redis OSS (mode cluster désactivé)
Ajouter des réplicas	Augmentation du nombre de réplicas dans une partition	Augmentation du nombre de réplicas dans une partition Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé)
Supprimer des réplicas	Diminution du nombre de réplicas dans une partition	Diminution du nombre de réplicas dans une partition Suppression d'une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé)

Augmentation du nombre de réplicas dans une partition

Vous pouvez augmenter le nombre de répliques dans une partition Valkey ou Redis OSS (mode cluster activé) ou dans un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) jusqu'à un maximum de cinq. Vous pouvez le faire à l'aide de AWS Management Console, de AWS CLI, ou de l' ElastiCache API.

Rubriques

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

À l'aide du AWS Management Console

La procédure suivante utilise la console pour augmenter le nombre de répliques dans un groupe de réplication Valkey ou Redis OSS (mode cluster activé).

Pour augmenter le nombre de répliques dans les partitions

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS, puis choisissez le nom du groupe de réplication auquel vous souhaitez ajouter des répliques.
3. Cochez la case correspondant à chaque partition à laquelle vous souhaitez ajouter des réplicas.
4. Choisissez Add replicas (Ajouter des réplicas).
5. Renseignez la page Add Replicas to Shards (Ajouter des réplicas à des partitions) :
   - Dans New number of replicas/shard (Nouveau nombre de réplicas par partition), saisissez le nombre de réplicas que doivent avoir chacune des partitions sélectionnées. Cette valeur doit être supérieure ou égale à la valeur de Current Number of Replicas per shard (Nombre actuel de réplicas par partition), et inférieure ou égale à cinq. Nous vous recommandons d'utiliser au moins deux réplicas.
   - Pour les zones de disponibilité, sélectionnez Aucune préférence pour avoir ElastiCache choisi une zone de disponibilité pour chaque nouvelle réplique, ou Spécifier les zones de disponibilité pour choisir une zone de disponibilité pour chaque nouvelle réplique.

Si vous choisissez Specify Availability Zones (Spécifier les zones de disponibilité), spécifiez une zone de disponibilité à l'aide de la liste pour chaque nouveau réplica.

6. Choisissez Add (Ajouter) pour ajouter les réplicas ou Cancel (Annuler) pour annuler l'opération.

À l'aide du AWS CLI

Pour augmenter le nombre de répliques dans une partition Valkey ou Redis OSS, utilisez la `increase-replica-count` commande avec les paramètres suivants :

- `--replication-group-id` – Obligatoire. Identifie le groupe de réplication dans lequel vous souhaitez augmenter le nombre de réplicas.
- `--apply-immediately` ou `--no-apply-immediately` – Obligatoire. Spécifie si le nombre de réplicas doit être augmenté immédiatement (`--apply-immediately`) ou lors du prochain créneau de maintenance (`--no-apply-immediately`). Actuellement, `--no-apply-immediately` n'est pas pris en charge.
- `--new-replica-count` : facultatif. Spécifie le nombre de nœuds de réplica que vous souhaitez avoir une fois l'opération terminée, dans la limite de 5 maximum. Utilisez ce paramètre pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) où il n'existe qu'un seul groupe de nœuds ou pour les groupes Valkey ou Redis OSS (mode cluster activé), ou pour lesquels vous souhaitez que tous les groupes de nœuds aient le même nombre de répliques. Si cette valeur n'est pas supérieure au nombre de réplicas actuel du groupe de nœuds, l'appel échoue avec une exception.
- `--replica-configuration` : facultatif. Vous permet de définir le nombre de réplicas et de zones de disponibilité pour chaque groupe de nœuds indépendamment. Utilisez ce paramètre pour les groupes Valkey ou Redis OSS (mode cluster activé) dans lesquels vous souhaitez configurer chaque groupe de nœuds indépendamment.

  `--replica-configuration` comporte trois chiffre facultatifs :

  - `NodeGroupId` : ID de quatre chiffres pour le groupe de nœud que vous configurez. Pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé), l'ID de partition est toujours. `0001` Pour trouver l'ID d'un groupe de nœuds (partition) Valkey ou Redis OSS (mode cluster activé), consultez. Recherche de l'ID d'une partition
  - `NewReplicaCount` – Nombre de réplicas que vous souhaitez dans ce groupe de nœuds à la fin de cette opération. La valeur doit être supérieure au nombre actuel de réplicas, dans la limite de

cinq maximum. Si cette valeur n'est pas supérieure au nombre de réplicas actuel du groupe de nœuds, l'appel échoue avec une exception.

- `PreferredAvailabilityZones` – Liste de chaînes `PreferredAvailabilityZone` qui spécifie dans quelles zones de disponibilité les nœuds des groupes de réplication doivent se trouver. Le nombre de valeurs `PreferredAvailabilityZone` doit être égal à la valeur de `NewReplicaCount` plus 1 afin de prendre en compte le nœud principal. Si ce membre de `--replica-configuration` est omis, ElastiCache pour Redis OSS, il choisit la zone de disponibilité pour chacune des nouvelles répliques.

> **Important**
>
> Vous devez inclure le paramètre `--new-replica-count` ou le paramètre `--replica-configuration`, mais pas les deux, dans votre appel.

Example

L'exemple suivant fait passer à 3 le nombre de réplicas du groupe de réplication `sample-repl-group`. À la fin de cet exemple, il y a trois réplicas dans chaque groupe de nœuds. Ce numéro s'applique qu'il s'agisse d'un groupe Valkey ou Redis OSS (mode cluster désactivé) avec un seul groupe de nœuds ou d'un groupe Valkey ou Redis OSS (mode cluster activé) avec plusieurs groupes de nœuds.

Pour Linux, macOS ou Unix :

```
aws elasticache increase-replica-count \
    --replication-group-id sample-repl-group \
    --new-replica-count 3 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache increase-replica-count ^
    --replication-group-id sample-repl-group ^
    --new-replica-count 3 ^
    --apply-immediately
```

L'exemple suivant fait passer le nombre de réplicas du groupe de réplication `sample-repl-group` à la valeur spécifiée pour les deux groupes de nœuds spécifiés. Étant donné qu'il existe plusieurs groupes de nœuds, il s'agit d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Lorsque vous spécifiez le paramètre `PreferredAvailabilityZones` facultatif, le nombre de zones de disponibilité indiqué doit correspondre à la valeur de `NewReplicaCount` plus 1. Cette approche prend en compte le nœud principal pour le groupe identifié par `NodeGroupId`.

Pour Linux, macOS ou Unix :

```
aws elasticache increase-replica-count \
    --replication-group-id sample-repl-group \
    --replica-configuration \
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
    --apply-immediately
```

Pour Windows :

```
aws elasticache increase-replica-count ^
    --replication-group-id sample-repl-group ^
    --replica-configuration ^
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \
    --apply-immediately
```

Pour plus d'informations sur l'augmentation du nombre de répliques à l'aide de la CLI, consultez le increase-replica-countmanuel Amazon ElastiCache Command Line Reference.

Utilisation de l' ElastiCache API

Pour augmenter le nombre de répliques dans une partition Valkey ou Redis OSS, utilisez l'`IncreaseReplicaCount`action avec les paramètres suivants :

- `ReplicationGroupId` – Obligatoire. Identifie le groupe de réplication dans lequel vous souhaitez augmenter le nombre de réplicas.
- `ApplyImmediately` – Obligatoire. Spécifie si le nombre de réplicas doit être augmenté immédiatement (`ApplyImmediately=True`) ou lors du prochain créneau de maintenance

(`ApplyImmediately=False`). Actuellement, `ApplyImmediately=False` n'est pas pris en charge.

- `NewReplicaCount` : facultatif. Spécifie le nombre de nœuds de réplica que vous souhaitez avoir une fois l'opération terminée, dans la limite de 5 maximum. Utilisez ce paramètre pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé) où il n'existe qu'un seul groupe de nœuds, ou pour les groupes Valkey ou Redis OSS (mode cluster activé) où vous souhaitez que tous les groupes de nœuds aient le même nombre de répliques. Si cette valeur n'est pas supérieure au nombre de réplicas actuel du groupe de nœuds, l'appel échoue avec une exception.
- `ReplicaConfiguration` : facultatif. Vous permet de définir le nombre de réplicas et de zones de disponibilité pour chaque groupe de nœuds indépendamment. Utilisez ce paramètre pour les groupes Valkey ou Redis OSS (mode cluster activé) dans lesquels vous souhaitez configurer chaque groupe de nœuds indépendamment.

  `ReplicaConfiguraion` comporte trois chiffre facultatifs :

  - `NodeGroupId` : ID de quatre chiffres pour le groupe de nœud que vous configurez. Pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé), l'ID du groupe de nœuds (partition) est toujours. `0001` Pour trouver l'ID d'un groupe de nœuds (partition) Valkey ou Redis OSS (mode cluster activé), consultez. Recherche de l'ID d'une partition
  - `NewReplicaCount` – Nombre de réplicas que vous souhaitez dans ce groupe de nœuds à la fin de cette opération. La valeur doit être supérieure au nombre actuel de réplicas, dans la limite de cinq maximum. Si cette valeur n'est pas supérieure au nombre de réplicas actuel du groupe de nœuds, l'appel échoue avec une exception.
  - `PreferredAvailabilityZones` – Liste de chaînes `PreferredAvailabilityZone` qui spécifie dans quelles zones de disponibilité les nœuds des groupes de réplication doivent se trouver. Le nombre de valeurs `PreferredAvailabilityZone` doit être égal à la valeur de `NewReplicaCount` plus 1 afin de prendre en compte le nœud principal. Si ce membre de `ReplicaConfiguration` est omis, ElastiCache pour Redis OSS, il choisit la zone de disponibilité pour chacune des nouvelles répliques.

> **Important**
>
> Vous devez inclure le paramètre `NewReplicaCount` ou le paramètre `ReplicaConfiguration`, mais pas les deux, dans votre appel.

Example

L'exemple suivant fait passer à 3 le nombre de réplicas du groupe de réplication `sample-repl-group`. À la fin de cet exemple, il y a trois réplicas dans chaque groupe de nœuds. Ce numéro s'applique qu'il s'agisse d'un groupe Valkey ou Redis OSS (mode cluster désactivé) avec un seul groupe de nœuds, ou d'un groupe Valkey ou Redis OSS (mode cluster activé) avec plusieurs groupes de nœuds.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=IncreaseReplicaCount
      &ApplyImmediately=True
      &NewReplicaCount=3
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

L'exemple suivant fait passer le nombre de réplicas du groupe de réplication `sample-repl-group` à la valeur spécifiée pour les deux groupes de nœuds spécifiés. Étant donné qu'il existe plusieurs groupes de nœuds, il s'agit d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Lorsque vous spécifiez le paramètre `PreferredAvailabilityZones` facultatif, le nombre de zones de disponibilité indiqué doit correspondre à la valeur de `NewReplicaCount` plus 1. Cette approche prend en compte le nœud principal pour le groupe identifié par `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=IncreaseReplicaCount
      &ApplyImmediately=True
      &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
      &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1b
      &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
```

```
      &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Pour plus d'informations sur l'augmentation du nombre de répliques à l'aide de l'API, consultez IncreaseReplicaCountle Amazon ElastiCache API Reference.

Diminution du nombre de réplicas dans une partition

Vous pouvez réduire le nombre de répliques dans une partition pour Valkey ou Redis OSS (mode cluster activé), ou dans un groupe de réplication pour Valkey ou Redis OSS (mode cluster désactivé) :

- Pour Valkey ou Redis OSS (mode cluster désactivé), vous pouvez réduire le nombre de répliques à un si le mode multi-AZ est activé, et à zéro s'il n'est pas activé.
- Pour Valkey ou Redis OSS (mode cluster activé), vous pouvez réduire le nombre de répliques à zéro. Cependant, vous ne pouvez pas basculer vers un réplica en cas d'échec du nœud principal.

Vous pouvez utiliser l' AWS Management Console API AWS CLI ou l' ElastiCache API pour réduire le nombre de répliques dans un groupe de nœuds (partition) ou un groupe de réplication.

Rubriques

- En utilisant le AWS Management Console
- En utilisant le AWS CLI
- Utilisation de l' ElastiCache API

En utilisant le AWS Management Console

La procédure suivante utilise la console pour diminuer le nombre de répliques dans un groupe de réplication Valkey ou Redis OSS (mode cluster activé).

Pour réduire le nombre de répliques dans une partition Valkey ou Redis OSS

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS, puis choisissez le nom du groupe de réplication dont vous souhaitez supprimer les répliques.
3. Cochez la case correspondant à chaque partition dont vous souhaitez supprimer un nœud de réplica.
4. Choisissez Delete replicas (Supprimer des réplicas).
5. Renseignez la page Delete Replicas from Shards (Supprimer des réplicas de partitions) :

    a. Dans New number of replicas/shard (Nouveau nombre de réplicas par partition), saisissez le nombre de réplicas que doivent avoir chacune des partitions sélectionnées. Cette valeur doit

être supérieure ou égale à 1. Nous vous recommandons d'utiliser au moins deux réplicas par partition.

b. Choisissez Delete (Supprimer) pour supprimer les réplicas ou Cancel (Annuler) pour annuler l'opération.

> **Important**
>
> - Si vous ne spécifiez pas les nœuds de réplication à supprimer, ElastiCache Redis OSS sélectionne automatiquement les nœuds de réplication à supprimer. Ce faisant, ElastiCache Redis OSS tente de conserver l'architecture multi-AZ de votre groupe de réplication, puis de conserver les répliques avec un délai de réplication minimal par rapport au principal.
> - Vous ne pouvez pas supprimer le nœud primaire ou le nœud maître dans un groupe de réplication. Si vous spécifiez un nœud principal afin qu'il soit supprimé, l'opération échoue et une erreur indique que le nœud principal a été sélectionné pour être supprimé.

En utilisant le AWS CLI

Pour réduire le nombre de répliques dans une partition Valkey ou Redis OSS, utilisez la `decrease-replica-count` commande avec les paramètres suivants :

- `--replication-group-id` – Obligatoire. Identifie le groupe de réplication dans lequel vous souhaitez diminuer le nombre de réplicas.
- `--apply-immediately` ou `--no-apply-immediately` : obligatoire. Spécifie si le nombre de réplicas doit être diminué immédiatement (`--apply-immediately`) ou lors du prochain créneau de maintenance (`--no-apply-immediately`). Actuellement, `--no-apply-immediately` n'est pas pris en charge.
- `--new-replica-count` : facultatif. Spécifie le nombre de nœuds de réplica souhaité. La valeur de `--new-replica-count` doit être une valeur valide inférieure au nombre de réplicas actuel dans les groupes de nœuds. Pour consulter les valeurs minimum autorisées, consultez Diminution du nombre de réplicas dans une partition. Si la valeur de `--new-replica-count` ne répond pas à cette exigence, l'appel échoue.
- `--replicas-to-remove` : facultatif. Contient une liste de nœuds IDs spécifiant les nœuds répliques à supprimer.

- `--replica-configuration` : facultatif. Vous permet de définir le nombre de réplicas et de zones de disponibilité pour chaque groupe de nœuds indépendamment. Utilisez ce paramètre pour les groupes Valkey ou Redis OSS (mode cluster activé) dans lesquels vous souhaitez configurer chaque groupe de nœuds indépendamment.

  `--replica-configuration` comporte trois chiffre facultatifs :

  - `NodeGroupId` : ID de quatre chiffres pour le groupe de nœud que vous configurez. Pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé), l'ID de partition est toujours. `0001` Pour trouver l'ID d'un groupe de nœuds (partition) Valkey ou Redis OSS (mode cluster activé), consultez. Recherche de l'ID d'une partition
  - `NewReplicaCount` : paramètre facultatif qui spécifie le nombre de nœuds de réplica souhaité. La valeur de `NewReplicaCount` doit être une valeur valide inférieure au nombre de réplicas actuel dans les groupes de nœuds. Pour consulter les valeurs minimum autorisées, consultez Diminution du nombre de réplicas dans une partition. Si la valeur de `NewReplicaCount` ne répond pas à cette exigence, l'appel échoue.
  - `PreferredAvailabilityZones` : liste de chaînes `PreferredAvailabilityZone` qui spécifie dans quelles zones de disponibilité les nœuds des groupes de réplication doivent se trouver. Le nombre de valeurs `PreferredAvailabilityZone` doit être égal à la valeur de `NewReplicaCount` plus 1 afin de prendre en compte le nœud principal. Si ce membre de `--replica-configuration` est omis, ElastiCache pour Redis OSS, il choisit la zone de disponibilité pour chacune des nouvelles répliques.

> **Important**
>
> Vous devez uniquement inclure l'un des paramètres suivants : `--new-replica-count`, `--replicas-to-remove` ou `--replica-configuration`.

Example

L'exemple suivant utilise `--new-replica-count` pour abaisser à 1 le nombre de réplicas du groupe de réplication `sample-repl-group`. À la fin de cet exemple, il y a un réplica dans chaque groupe de nœuds. Ce numéro s'applique qu'il s'agisse d'un groupe Valkey ou Redis OSS (mode cluster désactivé) avec un seul groupe de nœuds ou d'un groupe Valkey ou Redis OSS (mode cluster activé) avec plusieurs groupes de nœuds.

Pour Linux, macOS ou Unix :

```
aws elasticache decrease-replica-count
    --replication-group-id sample-repl-group \
    --new-replica-count 1 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --new-replica-count 1 ^
    --apply-immediately
```

L'exemple suivant diminue le nombre de réplicas du groupe de réplication `sample-repl-group` en supprimant deux réplicas spécifiés (`0001` et `0003`) du groupe de nœuds.

Pour Linux, macOS ou Unix :

```
aws elasticache decrease-replica-count \
    --replication-group-id sample-repl-group \
    --replicas-to-remove 0001,0003 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --replicas-to-remove 0001,0003 \
    --apply-immediately
```

L'exemple suivant utilise `--replica-configuration` pour diminuer le nombre de réplicas du groupe de réplication `sample-repl-group` à la valeur spécifiée pour les deux groupes de nœuds spécifiés. Étant donné qu'il existe plusieurs groupes de nœuds, il s'agit d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Lorsque vous spécifiez le paramètre `PreferredAvailabilityZones` facultatif, le nombre de zones de disponibilité indiqué doit correspondre à la valeur de `NewReplicaCount` plus 1. Cette approche prend en compte le nœud principal pour le groupe identifié par `NodeGroupId`.

Pour Linux, macOS ou Unix :

```
aws elasticache decrease-replica-count \
```

```
    --replication-group-id sample-repl-group \
    --replica-configuration \
        NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-east-1c \
        NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
    --apply-immediately
```

Pour Windows :

```
aws elasticache decrease-replica-count ^
    --replication-group-id sample-repl-group ^
    --replica-configuration ^
        NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c ^
        NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
    --apply-immediately
```

Pour plus d'informations sur la réduction du nombre de répliques à l'aide de la CLI, consultez le decrease-replica-countmanuel Amazon ElastiCache Command Line Reference.

Utilisation de l' ElastiCache API

Pour réduire le nombre de répliques dans une partition Valkey ou Redis OSS, utilisez l'`DecreaseReplicaCount`action avec les paramètres suivants :

- `ReplicationGroupId` – Obligatoire. Identifie le groupe de réplication dans lequel vous souhaitez diminuer le nombre de réplicas.
- `ApplyImmediately` : obligatoire. Spécifie si le nombre de réplicas doit être diminué immédiatement (`ApplyImmediately=True`) ou lors du prochain créneau de maintenance (`ApplyImmediately=False`). Actuellement, `ApplyImmediately=False` n'est pas pris en charge.
- `NewReplicaCount` : facultatif. Spécifie le nombre de nœuds de réplica souhaité. La valeur de `NewReplicaCount` doit être une valeur valide inférieure au nombre de réplicas actuel dans les groupes de nœuds. Pour consulter les valeurs minimum autorisées, consultez Diminution du nombre de réplicas dans une partition. Si la valeur de `--new-replica-count` ne répond pas à cette exigence, l'appel échoue.
- `ReplicasToRemove` : facultatif. Contient une liste de nœuds IDs spécifiant les nœuds répliques à supprimer.

- `ReplicaConfiguration` : facultatif. Contient une liste de groupes de nœuds qui vous permet de définir le nombre de réplicas et de zones de disponibilité pour chaque groupe de nœuds indépendamment. Utilisez ce paramètre pour les groupes Valkey ou Redis OSS (mode cluster activé) dans lesquels vous souhaitez configurer chaque groupe de nœuds indépendamment.

  `ReplicaConfiguraion` comporte trois chiffre facultatifs :

  - `NodeGroupId` : ID de quatre chiffres pour le groupe de nœud que vous configurez. Pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé), l'ID du groupe de nœuds est toujours. `0001` Pour trouver l'ID d'un groupe de nœuds (partition) Valkey ou Redis OSS (mode cluster activé), consultez. Recherche de l'ID d'une partition
  - `NewReplicaCount` : nombre de réplicas que vous souhaitez dans ce groupe de nœuds à la fin de cette opération. La valeur doit être inférieure au nombre actuel de réplicas et être de 1 minimum si l'option Multi-AZ est activée, ou de 0 si cette option n'est pas activée. Si cette valeur n'est pas inférieure au nombre de réplicas actuel du groupe de nœuds, l'appel échoue avec une exception.
  - `PreferredAvailabilityZones` : liste de chaînes `PreferredAvailabilityZone` qui spécifie dans quelles zones de disponibilité les nœuds des groupes de réplication doivent se trouver. Le nombre de valeurs `PreferredAvailabilityZone` doit être égal à la valeur de `NewReplicaCount` plus 1 afin de prendre en compte le nœud principal. Si ce membre de `ReplicaConfiguration` est omis, ElastiCache pour Redis OSS, il choisit la zone de disponibilité pour chacune des nouvelles répliques.

> **Important**
>
> Vous devez uniquement inclure l'un des paramètres suivants : `NewReplicaCount`, `ReplicasToRemove` ou `ReplicaConfiguration`.

Example

L'exemple suivant utilise `NewReplicaCount` pour abaisser à 1 le nombre de réplicas du groupe de réplication `sample-repl-group`. À la fin de cet exemple, il y a un réplica dans chaque groupe de nœuds. Ce numéro s'applique qu'il s'agisse d'un groupe Valkey ou Redis OSS (mode cluster désactivé) avec un seul groupe de nœuds ou d'un groupe Valkey ou Redis OSS (mode cluster activé) avec plusieurs groupes de nœuds.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &NewReplicaCount=1
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

L'exemple suivant diminue le nombre de réplicas du groupe de réplication `sample-repl-group` en supprimant deux réplicas spécifiés (`0001` et `0003`) du groupe de nœuds.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &ReplicasToRemove.ReplicaToRemove.1=0001
      &ReplicasToRemove.ReplicaToRemove.2=0003
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

L'exemple suivant utilise `ReplicaConfiguration` pour diminuer le nombre de réplicas du groupe de réplication `sample-repl-group` à la valeur spécifiée pour les deux groupes de nœuds spécifiés. Étant donné qu'il existe plusieurs groupes de nœuds, il s'agit d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé). Lorsque vous spécifiez le paramètre `PreferredAvailabilityZones` facultatif, le nombre de zones de disponibilité indiqué doit correspondre à la valeur de `NewReplicaCount` plus 1. Cette approche prend en compte le nœud principal pour le groupe identifié par `NodeGroupId`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=DecreaseReplicaCount
      &ApplyImmediately=True
      &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
      &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a
```

```
 &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
      &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
      &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

 &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
      &ReplicationGroupId=sample-repl-group
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Pour plus d'informations sur la réduction du nombre de répliques à l'aide de l'API, consultez DecreaseReplicaCountle Amazon ElastiCache API Reference.

Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé)

Les informations de la rubrique suivante s'appliquent uniquement aux groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).

À mesure que la charge de lecture augmente, vous pouvez avoir besoin de répartir ces lectures sur plusieurs nœuds et réduire la charge sur chaque nœud individuellement. Dans cette rubrique, vous découvrirez comment ajouter une réplique en lecture à un cluster Valkey ou Redis OSS (mode cluster désactivé).

Un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) peut avoir un maximum de cinq répliques en lecture. Si vous essayez d'ajouter un réplica en lecture à un groupe de réplication qui en contient déjà cinq, l'opération échoue.

Pour plus d'informations sur l'ajout de répliques à un groupe de réplication Valkey ou Redis OSS (mode cluster activé), consultez ce qui suit :

- Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

- Augmentation du nombre de réplicas dans une partition

Vous pouvez ajouter une réplique en lecture à un cluster Valkey ou Redis OSS (mode cluster désactivé) à l'aide de la ElastiCache console, de l'API ou de l' AWS CLI API. ElastiCache

Rubriques en relation

- Ajouter des nœuds à un ElastiCache cluster
- Ajout d'un réplica en lecture à un groupe de réplication (AWS CLI)
- Ajout d'un réplica en lecture à un groupe de réplication à l'aide de l'API

Ajout d'un réplica en lecture à un groupe de réplication (AWS CLI)

Pour ajouter une réplique en lecture à un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé), utilisez la AWS CLI `create-cache-cluster` commande, avec le paramètre `--replication-group-id` pour spécifier le groupe de réplication auquel ajouter le cluster (nœud).

L'exemple suivant crée le cluster `my-read replica` et l'ajoute au groupe de réplication `my-replication-group`. Les types de nœud, les groupes de paramètres, les groupes de sécurité, le créneau de maintenance et les autres paramètres du réplica en lecture seront les mêmes que ceux des autres nœuds dans `my-replication-group`.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-cluster \
      --cache-cluster-id my-read-replica \
      --replication-group-id my-replication-group
```

Pour Windows :

```
aws elasticache create-cache-cluster ^
      --cache-cluster-id my-read-replica ^
      --replication-group-id my-replication-group
```

Pour plus d'informations sur l'ajout d'une réplique en lecture à l'aide de la CLI, consultez create-cache-cluster le manuel Amazon ElastiCache Command Line Reference.

Ajout d'un réplica en lecture à un groupe de réplication à l'aide de l'API

Pour ajouter une réplique en lecture à un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé), utilisez l' ElastiCache `CreateCacheCluster`opération, avec le paramètre `ReplicationGroupId` pour spécifier le groupe de réplication auquel ajouter le cluster (nœud).

L'exemple suivant crée le cluster `myReadReplica` et l'ajoute au groupe de réplication `myReplicationGroup`. Les types de nœud, les groupes de paramètres, les groupes de sécurité, le créneau de maintenance et les autres paramètres du réplica en lecture seront les mêmes que ceux des autres nœuds `myReplicationGroup`.

```
https://elasticache.us-west-2.amazonaws.com/
      ?Action=CreateCacheCluster
      &CacheClusterId=myReadReplica
      &ReplicationGroupId=myReplicationGroup
      &Version=2015-02-02
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20150202T192317Z
      &X-Amz-Credential=<credential>
```

Pour plus d'informations sur l'ajout d'une réplique en lecture à l'aide de l'API, consultez CreateCacheCluster le Amazon ElastiCache API Reference.

Suppression d'une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé)

Les informations de la rubrique suivante s'appliquent uniquement aux groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).

À mesure que le trafic de lecture sur votre groupe de réplication Valkey ou Redis OSS change, vous souhaiterez peut-être ajouter ou supprimer des répliques de lecture. La suppression d'un nœud d'un groupe de réplication équivaut à supprimer un cluster, avec les restrictions suivantes :

- Vous ne pouvez pas supprimer le réplica principal d'un groupe de réplication. Si vous souhaitez supprimer le réplica principal, procédez comme suit :

  1. Promouvez un réplica en lecture en réplica principal. Pour plus d'informations sur la promotion d'un réplica en lecture en réplica principal, consultez Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).
  2. Supprimez l'ancienne réplica principal. Reportez-vous au point suivant pour consulter une restriction à cette méthode.

- Si l'option Multi-AZ est activée sur un groupe de réplication, vous ne pouvez pas supprimer le dernier réplica en lecture de ce groupe de réplication. Dans ce cas, procédez comme suit :

  1. Modifiez le groupe de réplication en désactivant Multi-AZ. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.
  2. Supprimez le réplica en lecture.

Vous pouvez supprimer une réplique en lecture d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à l'aide de la ElastiCache console, du AWS CLI for ElastiCache ou de l' ElastiCache API.

Pour savoir comment supprimer un cluster d'un groupe de réplication Valkey ou Redis OSS, consultez ce qui suit :

- En utilisant le AWS Management Console
- Utilisation du AWS CLI pour supprimer un ElastiCache cluster
- Utilisation de l' ElastiCache API
- Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)
- Diminution du nombre de réplicas dans une partition

Promouvoir une réplique en lecture au statut principal, pour les groupes de réplication Valkey ou Redis OSS (mode cluster désactivé)

Les informations de la rubrique suivante s'appliquent uniquement aux groupes de réplication Valkey ou Redis OSS (mode cluster désactivé).

Vous pouvez promouvoir une réplique de lecture Valkey ou Redis OSS (mode cluster désactivé) en tant que réplique principale à l'aide de l'API AWS Management Console, de ou de l' ElastiCache API. AWS CLI Vous ne pouvez pas promouvoir un réplica en lecture en réplica principal lorsque l'option Multi-AZ avec basculement automatique est activée sur le groupe de réplication . Pour faire d'une réplique Valkey ou Redis OSS (mode cluster désactivé) une réplique principale sur un groupe de réplication compatible multi-AZ, procédez comme suit :

1. Modifiez le groupe de réplication pour désactiver le mode Multi-AZ (pour cela, tous les clusters ne doivent pas obligatoirement se trouver dans la même zone de disponibilité). Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.
2. Promouvez le réplica en lecture en réplica principal.
3. Modifiez le groupe de réplication pour réactiver Multi-AZ.

Le mode multi-AZ n'est pas disponible sur les groupes de réplication exécutant Redis OSS 2.6.13 ou une version antérieure.

À l'aide du AWS Management Console

La procédure suivante utilise la console pour promouvoir un nœud de réplica et nœud principal.

Pour promouvoir un réplica en lecture en réplica principal (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Si le réplica que vous souhaitez promouvoir est membre d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) dans lequel le mode multi-AZ est activé, modifiez le groupe de réplication pour désactiver le mode multi-AZ avant de continuer. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.
3. Choisissez Valkey ou Redis OSS, puis dans la liste des clusters, choisissez le groupe de réplication que vous souhaitez modifier. Ce groupe de réplication doit exécuter le moteur « Redis », et non le moteur « Clustered Redis », et doit avoir au moins 2 nœuds.

4. Dans la liste des nœuds, choisissez le nœud de réplica que vous souhaitez promouvoir en nœud principal, puis, dans Actions, choisissez Promote (Promouvoir).
5. Dans la boîte de dialogue Promote Read Replica (Promouvoir le réplica en lecture), exécutez l'une des actions suivantes :
   a. Pour Apply Immediately (Appliquer immédiatement), choisissez Yes (Oui) pour promouvoir immédiatement le réplica en lecture, ou No (Non) pour le promouvoir au créneau de maintenance suivant du cluster.
   b. Choisissez Promote pour promouvoir le réplica en lecture ou Cancel pour annuler l'opération.
6. Si l'option Multi-AZ est activée pour le cluster avant que vous ne commenciez le processus de promotion, patientez jusqu'à ce que le statut du groupe de réplication soit available (disponible), puis modifiez le cluster pour activer à nouveau Multi-AZ. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.

À l'aide du AWS CLI

Il est impossible de promouvoir un réplica en lecture en réplica principal si Multi-AZ est activé pour le groupe de réplication. Dans certains cas, le réplica que vous souhaitez promouvoir peut appartenir à un groupe de réplication pour lequel le mode Multi-AZ est activé. Dans ces cas, vous devez modifier le groupe de réplication pour pouvoir désactiver le mode Multi-AZ avant de continuer. Pour cela, tous les clusters ne doivent pas obligatoirement se trouver dans la même zone de disponibilité. Pour plus d'informations sur la modification d'un groupe de réplication, consultez Modification d'un groupe de réplication.

La AWS CLI commande suivante modifie le groupe de réplications`ample-repl-group`, faisant de la réplique en lecture `my-replica-1` le principal du groupe de réplication.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
   --replication-group-id sample-repl-group \
   --primary-cluster-id my-replica-1
```

Pour Windows :

```
aws elasticache modify-replication-group ^
   --replication-group-id sample-repl-group ^
```

```
    --primary-cluster-id my-replica-1
```

Pour plus d'informations sur la modification d'un groupe de réplication, consultez modify-replication-group le manuel Amazon ElastiCache Command Line Reference.

Utilisation de l' ElastiCache API

Il est impossible de promouvoir un réplica en lecture en réplica principal si Multi-AZ est activé pour le groupe de réplication. Dans certains cas, le réplica que vous souhaitez promouvoir peut appartenir à un groupe de réplication pour lequel le mode Multi-AZ est activé. Dans ces cas, vous devez modifier le groupe de réplication pour pouvoir désactiver le mode Multi-AZ avant de continuer. Pour cela, tous les clusters ne doivent pas obligatoirement se trouver dans la même zone de disponibilité. Pour plus d'informations sur la modification d'un groupe de réplication, consultez Modification d'un groupe de réplication.

L'action d' ElastiCache API suivante modifie le groupe de réplication`myReplGroup`, faisant de la réplique en lecture `myReplica-1` le principal du groupe de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyReplicationGroup
   &ReplicationGroupId=myReplGroup
   &PrimaryClusterId=myReplica-1
   &Version=2014-12-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20141201T220302Z
   &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
   &X-Amz-Date=20141201T220302Z
   &X-Amz-SignedHeaders=Host
   &X-Amz-Expires=20141201T220302Z
   &X-Amz-Credential=<credential>
   &X-Amz-Signature=<signature>
```

Pour plus d'informations sur la modification d'un groupe de réplication, consultez ModifyReplicationGroup le Amazon ElastiCache API Reference.

Gestion de la maintenance des ElastiCache clusters

Chaque cluster a un créneau de maintenance hebdomadaire au cours duquel toutes les modifications systèmes seront appliquées. Avec Valkey et Redis OSS, les groupes de réplication ont le même créneau de maintenance hebdomadaire. Si vous ne spécifiez pas de fenêtre de maintenance

préférée lorsque vous créez ou modifiez un cluster ou un groupe de réplication, ElastiCache assignez 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 Asie-Pacifique (Séoul)	12:00–20:00 UTC
`ap-northeast-3`	Région Asie-Pacifique (Osaka)	12:00–20:00 UTC
`ap-southeast-3`	Région Asie-Pacifique (Jakarta)	14:00–22:00 UTC
`ap-south-1`	Région Asie-Pacifique (Mumbai)	17:30–01:30 UTC
`ap-southeast-1`	Région Asie-Pacifique (Singapour)	14:00–22:00 UTC
`cn-north-1`	Région Chine (Beijing)	14:00–22:00 UTC
`cn-northwest-1`	Région Chine (Ningxia)	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
`eu-west-3`	Région Europe (Paris)	23:59–07:29 UTC
`af-south-1`	Région Afrique (Le Cap)	13:00–21:00 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

Code région	Nom de la région	Fenêtre de maintenance régionale
me-south-1	Middle East (Bahrain) Region	13:00–21:00 UTC
me-central-1	Région du Moyen-Orient (EAU)	13:00–21:00 UTC
eu-south-1	Europe (Milan) Region	21:00–05:00 UTC
sa-east-1	Région Amérique du Sud (São Paulo)	01:00–09:00 UTC
us-east-1	Région USA Est (Virginie du Nord)	03:00–11:00 UTC
us-east-2	Région USA Est (Ohio)	04:00–12:00 UTC
us-gov-west-1	AWS GovCloud (US) région	06:00–14:00 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 de la fenêtre de maintenance de votre cluster ou de votre groupe de réplication

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 ou groupe de réplication 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.

> **Note**
>
> Si vous souhaitez appliquer les modifications de type de nœud, mettez immédiatement à niveau le and/or moteur à l'aide de la AWS Management Console case « Appliquer maintenant ». Sinon, ces modifications seront appliquées au cours de votre prochaine fenêtre de maintenance planifiée. Pour utiliser l'API, consultez modify-replication-groupou modify-cache-cluster.

En savoir plus

Pour plus d'informations sur votre créneau de maintenance et le remplacement des nœuds, veuillez consulter :

- ElastiCache Maintenance : FAQ sur la maintenance et le remplacement des nœuds
- Remplacement de nœuds (Memcached)—Gestion du remplacement des nœuds pour Memcached
- Modification d'un ElastiCache cluster — Modification du créneau de maintenance d'un cluster
- Remplacement de nœuds (Valkey et Redis OSS) — Gestion du remplacement des nœuds
- Modification d'un groupe de réplication — Modification du créneau de maintenance d'un groupe de réplication

Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres

Amazon ElastiCache 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 les tableaux des paramètres Memcached, consultez. Paramètres spécifiques Memcached Pour les tableaux des paramètres Valkey et Redis OSS, consultez. Paramètres OSS Valkey et Redis

Bien entendu, certaines valeurs de paramètres telles que `maxmemory` sont déterminées par le type de nœud de et de moteur. Pour un tableau de ces valeurs de paramètres Memcached par type de nœud, consultez. Paramètres propres au type de nœud Memcached Pour un tableau de ces valeurs de paramètres Valkey et Redis OSS par type de nœud, consultez. Paramètres spécifiques au type de nœud Redis OSS

> **Note**
>
> Pour obtenir la liste des paramètres spécifiques à Memcached, consultez Paramètres spécifiques Memcached.

Rubriques

- Gestion des paramètres dans ElastiCache
- Le groupe de paramètres du cache intègre des niveaux ElastiCache
- Création d'un groupe ElastiCache de paramètres
- Répertorier les groupes de ElastiCache paramètres par nom

- Lister les valeurs d'un groupe de ElastiCache paramètres
- Modification d'un groupe ElastiCache de paramètres
- Supprimer un groupe ElastiCache de paramètres
- Paramètres spécifiques au moteur

Gestion des paramètres dans ElastiCache

ElastiCache les paramètres sont regroupés dans des groupes de paramètres nommés pour faciliter la gestion des paramètres. 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.
- Pour Memcached, la famille de groupes de paramètres et le cluster auquel vous l'attribuez doivent être compatibles. Par exemple, si votre cluster utilise Memcached version 1.4.8, vous pouvez utiliser uniquement des groupes de paramètres par défaut ou personnalisés de la famille Memcached 1.4.

  Pour Redis OSS, la famille de groupes de paramètres et le cluster auquel vous l'attribuez doivent être compatibles. Par exemple, si votre cluster exécute Redis OSS version 3.2.10, vous ne pouvez utiliser que des groupes de paramètres, par défaut ou personnalisés, de la famille Redis OSS 3.2.
- Si vous modifiez le groupe de paramètres d'un cluster, les valeurs de n'importe quel paramètre modifiable sous certaines conditions doivent être les mêmes dans les deux groupes de paramètre actuel et nouveau.
- Pour Memcached, lorsque vous modifiez les paramètres d'un cluster, les modifications sont immédiatement appliquées 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. Pour déterminer quand une modification de paramètre particulier sera appliquée, consultez la colonne Application des modifications dans les tableaux de Paramètres spécifiques Memcached. Pour en savoir plus sur le redémarrage des nœuds d'un cluster, consultez Redémarrage de clusters.
- Pour Redis OSS, lorsque vous modifiez les paramètres d'un cluster, la modification est appliquée au cluster soit immédiatement, soit, avec les exceptions indiquées ci-dessous, après le redémarrage des nœuds du 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. Pour déterminer

quand une modification de paramètre particulier sera appliquée, consultez la colonne Application des modifications dans les tableaux de Paramètres OSS Valkey et Redis.

Pour plus d'informations sur le redémarrage des nœuds Valkey ou Redis OSS, consultez. Redémarrage des nœuds

> Modifications des paramètres Valkey ou Redis OSS (mode cluster activé)
>
> Si vous modifiez les paramètres suivants sur un cluster Valkey ou Redis OSS (mode cluster activé), suivez les étapes suivantes.
>
> - activerehachage
> - bases de données
>
> 1. Créez une sauvegarde manuelle de votre cluster. Consultez Réalisation de sauvegardes manuelles.
> 2. Supprimez le cluster. Consultez Suppression de clusters.
> 3. Restaurez le cluster grâce au groupe de paramètres modifiés et à la sauvegarde pour alimenter le nouveau cluster. Consultez Restauration à partir d'une sauvegarde dans un nouveau cache.
>
> Les modifications apportées à d'autres paramètres ne nécessitent pas cela.

- Vous pouvez associer des groupes de paramètres aux banques de données globales Valkey et Redis OSS. Les banques de données globales sont un ensemble d'un ou de plusieurs clusters qui s'étendent sur AWS plusieurs régions. Dans ce cas, le groupe de paramètres est partagé par tous les clusters qui composent le magasin de données global. Toute modification apportée au groupe de paramètres du cluster principal est répliquée vers tous les autres clusters restants du magasin de données global. Pour de plus amples informations, veuillez consulter Réplication entre AWS régions à l'aide de banques de données mondiales.

  Vous pouvez vérifier si un groupe de paramètres fait partie d'un magasin de données global en effectuant une recherche aux emplacements suivants :

  - Sur la ElastiCache console, sur la page Groupes de paramètres, l'attribut yes/no Global
  - La yes/no `IsGlobal` propriété de l'opération CacheParameterGroupd'API

Le groupe de paramètres du cache intègre des niveaux ElastiCache

Amazon ElastiCache dispose de trois niveaux de groupes de paramètres de cache, comme indiqué ci-dessous.

Niveaux ElastiCache du groupe de paramètres Amazon

Par défaut global

Le groupe de paramètres racine de premier niveau pour tous les ElastiCache clients Amazon de la région.

Le groupe de paramètres de cache par défaut global :

- Est réservé au client ElastiCache et n'est pas disponible pour celui-ci.

Par défaut client

Une copie du groupe de paramètres de cache par défaut global qui est créée pour être utilisée par le client.

Le groupe de paramètres de cache par défaut client :

- Est créé et détenu par ElastiCache.
- Est disponible pour être utilisé par le client comme groupe de paramètres de cache pour tous les clusters qui exécutent une version de moteur prise en charge par ce groupe de paramètres de cache.

- Ne peut pas être modifié par le client.

Appartient au client

Une copie du groupe de paramètres de cache par défaut client. Un groupe de paramètres de cache appartenant au client est créé chaque fois que le client crée un groupe de paramètres de cache.

Le groupe de paramètres de cache par 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 de cache personnalisé.

  Toutes les valeurs de paramètre ne peuvent pas être modifiées. Pour plus d'informations sur les valeurs Memcached, consultez. Paramètres spécifiques Memcached Pour plus d'informations sur les valeurs Valkey et Redis OSS, consultez. Paramètres OSS Valkey et Redis

Création d'un groupe ElastiCache 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 ElastiCache console, de AWS CLI, ou de l' ElastiCache API.

Création d'un groupe de ElastiCache paramètres (console)

La procédure suivante montre comment créer un groupe de paramètres à l'aide de la console ElastiCache.

Pour créer un groupe de paramètres à l'aide de la ElastiCache console

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

   L'écran Create Parameter Group (Créer un groupe de paramètres) s'affiche.
4. A partir de la liste Famille, choisissez la famille groupe de paramètres qui sera le modèle de votre groupe de paramètres.

La famille de groupes de paramètres, telle que memcached1.4 ou redis3.2, définit les paramètres réels de votre groupe de paramètres ainsi que leurs valeurs initiales. La famille de groupe de paramètres doit correspondre avec le moteur du cluster et sa version.

5. 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.
6. Dans la zone Description, saisissez une description du groupe de paramètres.
7. Choisissez Créer pour créer le groupe de paramètres.

   Pour terminer le processus sans créer le groupe de paramètres, choisissez Annuler.
8. 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 de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

Création d'un groupe de ElastiCache paramètres (AWS CLI)

Pour créer un groupe de paramètres à l'aide de AWS CLI, utilisez la commande `create-cache-parameter-group` avec ces paramètres.

- `--cache-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.
- `--cache-parameter-group-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 paramètre nommé myMem14 à l'aide de la famille memcached1.4 comme modèle.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-name myMem14  \
    --cache-parameter-group-family memcached1.4 \
    --description "My first parameter group"
```

Pour Windows :

```
aws elasticache create-cache-parameter-group ^
    --cache-parameter-group-name myMem14  ^
    --cache-parameter-group-family memcached1.4 ^
    --description "My first parameter group"
```

Le résultat de cette commande devrait ressembler à cet exemple.

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "myMem14",
        "CacheParameterGroupFamily": "memcached1.4",
        "Description": "My first  parameter group"
    }
}
```

Example

L'exemple suivant crée un paramètre nommé myRed28 à l'aide de la famille redis2.8 comme modèle.

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-parameter-group \
    --cache-parameter-group-name myRed28  \
    --cache-parameter-group-family redis2.8 \
    --description "My first parameter group"
```

Pour Windows :

```
aws elasticache create-cache-parameter-group ^
    --cache-parameter-group-name myRed28  ^
    --cache-parameter-group-family redis2.8 ^
    --description "My first parameter group"
```

Le résultat de cette commande devrait ressembler à cet exemple.

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "myRed28",
        "CacheParameterGroupFamily": "redis2.8",
        "Description": "My first parameter group"
    }
}
```

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 de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

Pour de plus amples informations, veuillez consulter `create-cache-parameter-group`.

Création d'un groupe de ElastiCache paramètres (ElastiCache API)

Pour créer un groupe de paramètres à l'aide de l' ElastiCache API, utilisez l'`CreateCacheParameterGroup`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.
- `CacheParameterGroupFamily` – La famille du moteur et de version pour le groupe de paramètres. Par exemple, `memcached1.4`.
- `CacheParameterGroupFamily` – La famille du moteur et de version pour le groupe de paramètres. Par exemple, `redis2.8`.
- `Description` – Une description fourni par l'utilisateur pour le groupe de paramètres.

Example

L'exemple suivant crée un paramètre nommé myMem14 à l'aide de la famille memcached1.4 comme modèle.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateCacheParameterGroup
   &CacheParameterGroupFamily=memcached1.4
   &CacheParameterGroupName=myMem14
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action devrait se présenter comme suit.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My first  parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Example

L'exemple suivant crée un paramètre nommé myRed28 à l'aide de la famille redis2.8 comme modèle.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=CreateCacheParameterGroup
   &CacheParameterGroupFamily=redis2.8
   &CacheParameterGroupName=myRed28
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action devrait se présenter comme suit.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
      <CacheParameterGroupName>myRed28</CacheParameterGroupName>
      <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
      <Description>My first parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

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 de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

Pour de plus amples informations, veuillez consulter `CreateCacheParameterGroup`.

Répertorier les groupes de ElastiCache paramètres par nom

Vous pouvez répertorier les groupes de paramètres à l'aide de la ElastiCache console, de AWS CLI, ou de l' ElastiCacheAPI.

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 ElastiCache .

Pour répertorier les groupes de paramètres à l'aide de la ElastiCache console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 ElastiCache paramètres par nom (AWS CLI)

Pour générer une liste de groupes de paramètres à l'aide de AWS CLI, utilisez la commande`describe-cache-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-records` 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 myMem14.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myMem14
```

Pour Windows :

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myMem14
```

Le résultat de cette commande se présentera de la façon suivante, avec le nom, la famille et la description du groupe de paramètres.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myMem14",
         "CacheParameterGroupFamily": "memcached1.4",
         "Description": "My first parameter group"
     }
    ]
}
```

Example

L'exemple de code suivant répertorie le groupe de paramètres myRed28.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myRed28
```

Pour Windows :

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myRed28
```

Le résultat de cette commande se présentera de la façon suivante, avec le nom, la famille et la description du groupe de paramètres.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myRed28",
         "CacheParameterGroupFamily": "redis2.8",
         "Description": "My first parameter group"
     }
    ]
}
```

Example

L'exemple de code suivant répertorie le groupe de paramètres MyRed56 pour les groupes de paramètres exécutés sur le moteur Redis OSS version 5.0.6 et ultérieure. Si le groupe de paramètres

fait partie d'un Réplication entre AWS régions à l'aide de banques de données mondiales, la valeur de la propriété `IsGlobal` renvoyée dans la sortie sera `Yes`.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name myRed56
```

Pour Windows :

```
aws elasticache describe-cache-parameter-groups ^
    --cache-parameter-group-name myRed56
```

Le résultat de cette commande se présentera de la façon suivante, avec le nom, la famille, isGlobal et la description du groupe de paramètres.

```
{
    "CacheParameterGroups": [
     {
         "CacheParameterGroupName": "myRed56",
         "CacheParameterGroupFamily": "redis5.0",
         "Description": "My first parameter group",
         "IsGlobal": "yes"
     }
    ]
}
```

Example

L'exemple de code suivant répertorie jusqu'à 10 groupes de paramètres.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

La sortie JSON de cette commande se présentera de la façon suivante, indiquant le nom, la famille, la description et, dans le cas de redis5.6, si le groupe de paramètres fait partie d'un entrepôt de données global (isGlobal), pour chaque groupe de paramètres.

```
{
    "CacheParameterGroups": [
```

```
        {
            "CacheParameterGroupName": "custom-redis32",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "custom parameter group with reserved-memory > 0"
        },
        {
            "CacheParameterGroupName": "default.memcached1.4",
            "CacheParameterGroupFamily": "memcached1.4",
            "Description": "Default parameter group for memcached1.4"
        },
        {
            "CacheParameterGroupName": "default.redis2.6",
            "CacheParameterGroupFamily": "redis2.6",
            "Description": "Default parameter group for redis2.6"
        },
        {
            "CacheParameterGroupName": "default.redis2.8",
            "CacheParameterGroupFamily": "redis2.8",
            "Description": "Default parameter group for redis2.8"
        },
        {
            "CacheParameterGroupName": "default.redis3.2",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "Default parameter group for redis3.2"
        },
        {
            "CacheParameterGroupName": "default.redis3.2.cluster.on",
            "CacheParameterGroupFamily": "redis3.2",
            "Description": "Customized default parameter group for redis3.2 with
 cluster mode on"
        },
        {
            "CacheParameterGroupName": "default.redis5.6.cluster.on",
            "CacheParameterGroupFamily": "redis5.0",
            "Description": "Customized default parameter group for redis5.6 with
 cluster mode on",
            "isGlobal": "yes"
        },
    ]
}
```

Pour de plus amples informations, veuillez consulter `describe-cache-parameter-groups`.

Répertorier les groupes de ElastiCache paramètres par nom (ElastiCache API)

Pour générer une liste de groupes de paramètres à l'aide de l' ElastiCache API, utilisez l'`DescribeCacheParameterGroups`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 `MaxRecords` 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 myMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action se présentera de la façon suivante, avec le nom, la famille et la description de chaque groupe de paramètres.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

L'exemple de code suivant répertorie jusqu'à 10 groupes de paramètres.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &MaxRecords=10
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à cette action se présentera de la façon suivante, indiquant le nom, la famille, la description et, dans le cas de redis5.6, si le groupe de paramètres appartient à un entrepôt de données global (isGlobal), pour chaque groupe de paramètres.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
       <CacheParameterGroup>
        <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
```

```
</DescribeCacheParameterGroupsResponse>
```

Example

L'exemple de code suivant répertorie le groupe de paramètres myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myRed28
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à cette action se présentera de la façon suivante, avec le nom, la famille et la description.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

L'exemple de code suivant répertorie le groupe de paramètres myRed56.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameterGroups
   &CacheParameterGroupName=myRed56
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
```

```
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

La réponse à cette action se présentera de la façon suivante, indiquant le nom, la famille, la description et si le groupe de paramètres fait partie d'un entrepôt de données global (isGlobal).

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Pour de plus amples informations, veuillez consulter DescribeCacheParameterGroups.

Lister les valeurs d'un groupe de ElastiCache paramètres

Vous pouvez répertorier les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la ElastiCache console, de l' AWS CLI API ou de l' ElastiCache API.

Lister les valeurs d'un groupe de ElastiCache paramètres (console)

La procédure suivante indique comment répertorier les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la ElastiCache console.

Pour afficher les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de la console ElastiCache

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 dont vous souhaitez afficher les paramètres et les valeurs en cochant la case située à gauche 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.

Affichage des 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-cache-parameters`.

Example

L'exemple de code suivant répertorie tous les paramètres Memcached et leurs valeurs pour le groupe de paramètres MyMem14.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name myMem14
```

Pour Windows :

```
aws elasticache describe-cache-parameters ^
    --cache-parameter-group-name myMem14
```

Example

L'exemple de code suivant affiche tous les paramètres et leurs valeurs pour le groupe de paramètres myRedis28.

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name myRedis28
```

Pour Windows :

```
aws elasticache describe-cache-parameters ^
    --cache-parameter-group-name myRed28
```

Pour de plus amples informations, veuillez consulter `describe-cache-parameters`.

Lister les valeurs d'un groupe de paramètres (ElastiCache API)

Pour répertorier les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de l' ElastiCache API, utilisez l'`DescribeCacheParameters`action.

Example

L'exemple de code suivant répertorie tous les paramètres Memcached pour le groupe de paramètres MyMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameters
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action devra se présenter comme suit. La réponse à cette a été tronquée.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
 in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>

...output omitted...

    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>
```

Example

L'exemple de code suivant affiche tous les paramètres du groupe de paramètres myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeCacheParameters
```

```
   &CacheParameterGroupName=myRed28
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

La réponse à partir de cette action devra se présenter comme suit. La réponse à cette a été tronquée.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to store items,
 in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>

...output omitted...

    </CacheClusterClassSpecificParameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
```

```
  </ResponseMetadata>
</DescribeCacheParametersResponse>
```

Pour de plus amples informations, veuillez consulter `DescribeCacheParameters`.

Modification d'un groupe ElastiCache 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 plus d'informations sur le moment où une modification de valeur de paramètre est appliquée à un groupe de paramètres, reportez-vous aux Paramètres OSS Valkey et Redis sections etParamètres spécifiques Memcached.

Modification d'un groupe de paramètres

La procédure suivante montre comment modifier la valeur du `cluster-enabled` paramètre à l'aide de la ElastiCache console. 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 ElastiCache console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 à modifier en cochant la case située à gauche du nom du groupe de paramètres.

   Les paramètres du groupe de paramètres figureront au bas de l'écran. Vous devrez peut-être parcourir la liste pour afficher tous les paramètres.
4. Pour modifier un ou plusieurs paramètres, choisissez Modifier les paramètres.
5. Sur l'écran Edit Parameter Group: faites défiler, à l'aide des flèches gauche et droite, pour rechercher le paramètre `binding_protocol`, puis tapez `ascii` dans la colonne Value.

6. Choisissez Save Changes (Enregistrer les modifications).
7. Pour Memcached, pour trouver le nom du paramètre que vous avez modifié, consultez. Paramètres spécifiques Memcached Si des modifications apportées au paramètre prennent effet après le redémarrage, redémarrez chaque cluster qui utilise ce groupe de paramètres. Pour plus d'informations, consultez Redémarrage de clusters.
8. Avec Valkey et Redis OSS, pour trouver le nom du paramètre que vous avez modifié, consultez. Paramètres OSS Valkey et Redis Si vous avez un cluster Valkey ou Redis OSS (mode cluster désactivé) et que vous modifiez les paramètres suivants, vous devez redémarrer les nœuds du cluster :

   - activerehachage
   - bases de données

   Pour plus d'informations sur le redémarrage, veuillez consulter Redémarrage des nœuds.

   > **Modifications des paramètres Valkey ou Redis OSS (mode cluster activé)**
   >
   > Si vous modifiez les paramètres suivants sur un cluster Valkey ou Redis OSS (mode cluster activé), suivez les étapes suivantes.
   >
   > - activerehachage
   > - bases de données
   >
   > 1. Avec Redis OSS, vous pouvez créer une sauvegarde manuelle de votre cluster. Consultez Réalisation de sauvegardes manuelles.
   > 2. Supprimez le cluster. Consultez Suppression de clusters.
   > 3. Restaurez le cluster grâce au groupe de paramètres modifiés et à la sauvegarde pour alimenter le nouveau cluster. Consultez Restauration à partir d'une sauvegarde dans un nouveau cache.
   >
   > Les modifications apportées à d'autres paramètres ne nécessitent pas cela.

Modification d'un groupe de paramètres (AWS CLI)

Pour modifier la valeur d'un paramètre à l'aide de AWS CLI, utilisez la commande`modify-cache-parameter-group`.

Example

Avec Memcached, pour trouver le nom et les valeurs autorisées du paramètre que vous souhaitez modifier, consultez Paramètres spécifiques Memcached

L'exemple de code suivant définir la valeur de deux paramètres, chunk_size et chunk_size_growth_fact sur le groupe de paramètres `myMem14`.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name myMem14 \
    --parameter-name-values \
        ParameterName=chunk_size,ParameterValue=96 \
        ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Pour Windows :

```
aws elasticache modify-cache-parameter-group ^
    --cache-parameter-group-name myMem14 ^
    --parameter-name-values ^
        ParameterName=chunk_size,ParameterValue=96 ^
        ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Le résultat de cette commande doit être semblable à ce qui suit.

```
{
    "CacheParameterGroupName": "myMem14"
}
```

Example

Avec Valkey et Redis OSS, pour trouver le nom et les valeurs autorisées du paramètre que vous souhaitez modifier, voir Paramètres OSS Valkey et Redis

L'exemple de code suivant définit la valeur de deux paramètres reserved-memory-percentet active le cluster sur le groupe de paramètres. `myredis32-on-30` Nous avons réglé reserved-memory-

percentsur `30` (30 %) et activé le cluster sur `yes` afin que le groupe de paramètres puisse être utilisé avec les clusters Valkey ou Redis OSS (mode cluster activé) (groupes de réplication).

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name myredis32-on-30 \
    --parameter-name-values \
        ParameterName=reserved-memory-percent,ParameterValue=30 \
        ParameterName=cluster-enabled,ParameterValue=yes
```

Pour Windows :

```
aws elasticache modify-cache-parameter-group ^
    --cache-parameter-group-name myredis32-on-30 ^
    --parameter-name-values ^
        ParameterName=reserved-memory-percent,ParameterValue=30 ^
        ParameterName=cluster-enabled,ParameterValue=yes
```

Le résultat de cette commande doit être semblable à ce qui suit.

```
{
    "CacheParameterGroupName": "my-redis32-on-30"
}
```

Pour de plus amples informations, veuillez consulter `modify-cache-parameter-group`.

Pour rechercher le nom du paramètre que vous avez modifié, consultez Paramètres OSS Valkey et Redis.

Si vous avez un cluster Valkey ou Redis OSS (mode cluster désactivé) et que vous modifiez les paramètres suivants, vous devez redémarrer les nœuds du cluster :

- activerehachage
- bases de données

Pour plus d'informations sur le redémarrage, veuillez consulter Redémarrage des nœuds.

Modifications des paramètres Valkey ou Redis OSS (mode cluster activé)

Si vous modifiez les paramètres suivants sur un cluster Valkey ou Redis OSS (mode cluster activé), suivez les étapes suivantes.

- activerehachage
- bases de données

1. Créez une sauvegarde manuelle de votre cluster. Consultez Réalisation de sauvegardes manuelles.
2. Supprimez le cluster. Consultez Suppression de clusters.
3. Restaurez le cluster grâce au groupe de paramètres modifiés et à la sauvegarde pour alimenter le nouveau cluster. Consultez Restauration à partir d'une sauvegarde dans un nouveau cache.

Les modifications apportées à d'autres paramètres ne nécessitent pas cela.

Modification d'un groupe de paramètres (ElastiCache API)

Pour modifier les valeurs des paramètres d'un groupe de paramètres à l'aide de l' ElastiCache API, utilisez l'`ModifyCacheParameterGroup`action.

Example

Avec Memcached, pour trouver le nom et les valeurs autorisées du paramètre que vous souhaitez modifier, consultez Paramètres spécifiques Memcached

L'exemple de code suivant définir la valeur de deux paramètres, chunk_size et chunk_size_growth_fact sur le groupe de paramètres `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyCacheParameterGroup
   &CacheParameterGroupName=myMem14
   &ParameterNameValues.member.1.ParameterName=chunk_size
   &ParameterNameValues.member.1.ParameterValue=96
   &ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
   &ParameterNameValues.member.2.ParameterValue=1.5
```

```
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Example

Avec Valkey et Redis OSS, pour trouver le nom et les valeurs autorisées du paramètre que vous souhaitez modifier, voir Paramètres OSS Valkey et Redis

L'exemple de code suivant définit la valeur de deux paramètres reserved-memory-percentet active le cluster sur le groupe de paramètres. `myredis32-on-30` Nous avons réglé reserved-memory-percentsur `30` (30 %) et activé le cluster sur `yes` afin que le groupe de paramètres puisse être utilisé avec les clusters Valkey ou Redis OSS (mode cluster activé) (groupes de réplication).

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ModifyCacheParameterGroup
   &CacheParameterGroupName=myredis32-on-30
   &ParameterNameValues.member.1.ParameterName=reserved-memory-percent
   &ParameterNameValues.member.1.ParameterValue=30
   &ParameterNameValues.member.2.ParameterName=cluster-enabled
   &ParameterNameValues.member.2.ParameterValue=yes
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Pour de plus amples informations, veuillez consulter `ModifyCacheParameterGroup`.

Si vous avez un cluster Valkey ou Redis OSS (mode cluster désactivé) et que vous modifiez les paramètres suivants, vous devez redémarrer les nœuds du cluster :

- activerehachage
- bases de données

Pour de plus amples informations, veuillez consulter Redémarrage des nœuds.

Modifications des paramètres Valkey ou Redis OSS (mode cluster activé)

Si vous modifiez les paramètres suivants sur un cluster Valkey ou Redis OSS (mode cluster activé), suivez les étapes suivantes.

- activerehachage
- bases de données

1. Créez une sauvegarde manuelle de votre cluster. Consultez Réalisation de sauvegardes manuelles.
2. Supprimez le cluster. Consultez Supprimer un cluster dans ElastiCache.
3. Restaurez le cluster grâce au groupe de paramètres modifiés et à la sauvegarde pour alimenter le nouveau cluster. Consultez Restauration à partir d'une sauvegarde dans un nouveau cache.

Les modifications apportées à d'autres paramètres ne nécessitent pas cela.

Supprimer un groupe ElastiCache de paramètres

Vous pouvez supprimer un groupe de paramètres personnalisé à l'aide de la ElastiCache console, de AWS CLI, ou de l'ElastiCache API.

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 ElastiCache.

Pour supprimer un groupe de paramètres à l'aide de la ElastiCache console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 à supprimer en cochant la case située à gauche du nom du groupe de paramètres.

   Le bouton Supprimer devient actif.
4. Sélectionnez Delete (Supprimer).

   L'écran de confirmation Delete Parameter Groups s'affichera.
5. Pour supprimer les groupes de paramètres, sur l'écran de confirmation Supprimer les groupes de paramètres , choisissez Supprimer.

   Pour conserver les groupes de paramètres, choisissez Annuler.

Suppression d'un groupe de paramètres (AWS CLI)

Pour supprimer un groupe de paramètres à l'aide de AWS CLI, utilisez la commande`delete-cache-parameter-group`. Pour le groupe de paramètres à supprimer, le groupe de paramètres spécifié par `--cache-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 myMem14.

Example

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-parameter-group \
    --cache-parameter-group-name myRed28
```

Pour Windows :

```
aws elasticache delete-cache-parameter-group ^
    --cache-parameter-group-name myRed28
```

Pour de plus amples informations, veuillez consulter `delete-cache-parameter-group`.

Supprimer un groupe de paramètres (ElastiCache API)

Pour supprimer un groupe de paramètres à l'aide de l' ElastiCache API, utilisez l'`DeleteCacheParameterGroup`action. Pour le groupe de paramètres à supprimer, le groupe de paramètres spécifié par `CacheParameterGroupName` ne peut pas avoir de clusters associés, et ne peut pas être non plus un groupe de paramètres par défaut.

Example

Avec Memcached, l'exemple de code suivant supprime le groupe de paramètres MyMem14.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteCacheParameterGroup
   &CacheParameterGroupName=myMem14
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Example

L'exemple de code suivant supprime le groupe de paramètres myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DeleteCacheParameterGroup
   &CacheParameterGroupName=myRed28
```

```
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &Version=2015-02-02
    &X-Amz-Credential=<credential>
```

Pour de plus amples informations, veuillez consulter DeleteCacheParameterGroup.

Paramètres spécifiques au moteur

Valkey et Redis OSS

La plupart des paramètres Valkey 8 sont compatibles avec les paramètres Redis OSS 7.1. Les paramètres Valkey 7.2 sont les mêmes que ceux de Redis OSS 7.

Si vous ne spécifiez aucun groupe de paramètres pour votre cluster Valkey ou 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 de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

Rubriques

- Paramètres OSS Valkey et Redis
- Paramètres spécifiques Memcached

Paramètres OSS Valkey et Redis

Rubriques

- Modifications des paramètres de Valkey 8.1
- Modifications des paramètres de Valkey 8.0
- Modifications des paramètres de Valkey 7.2 et Redis OSS 7
- Modifications des paramètres de Redis OSS 6.x
- Modifications des paramètres de Redis OSS 5.0.3
- Modifications des paramètres de Redis OSS 5.0.0
- Modifications des paramètres de Redis OSS 4.0.10
- Modifications des paramètres de Redis OSS 3.2.10
- Modifications des paramètres de Redis OSS 3.2.6
- Modifications des paramètres de Redis OSS 3.2.4
- Paramètres ajoutés à Redis OSS 2.8.24 (amélioré)
- Paramètres ajoutés à Redis OSS 2.8.23 (amélioré)
- Paramètres ajoutés à Redis OSS 2.8.22 (amélioré)
- Paramètres ajoutés à Redis OSS 2.8.21
- Paramètres ajoutés à Redis OSS 2.8.19
- Paramètres ajoutés à Redis OSS 2.8.6
- Paramètres de Redis OSS 2.6.13
- Paramètres spécifiques au type de nœud Redis OSS

Modifications des paramètres de Valkey 8.1

Famille de groupes de paramètres : valkey8

> **Note**
>
> - Les modifications des paramètres de Valkey 8.1 ne s'appliquent pas à Valkey 8.0
> - Les groupes de paramètres Valkey 8.0 et versions ultérieures sont incompatibles avec Redis OSS 7.2.4.

- dans Valkey 8.1, les commandes suivantes ne sont pas disponibles pour les caches sans serveur :`commandlog`,,`commandlog get`, et `commandlog help` `commandlog len` `commandlog reset`.

Nouveaux groupes de paramètres dans Valkey 8.1

Nom	Détails	Description
commandlog-large-request-max-len (ajouté dans la version 8.1)	Par défaut: 1048576 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	La taille maximale, en octets, des requêtes à enregistrer par la fonctionnalité Valkey Command Log.
commandlog-large-request-max-len (ajouté dans la version 8.1)	Valeur par défaut : 128 Valeurs autorisées : 0 à 1024 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	La longueur maximale du journal des commandes Valkey pour les demandes.
commandlog-reply-larger-than (ajouté dans la version 8.1)	Par défaut: 1048576 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	Taille maximale, en octets, des réponses à enregistrer par la fonctionnalité Valkey Command Log.

Nom	Détails	Description
commandlog-large-reply-max-len (ajouté dans la version 8.1)	Valeur par défaut : 128 Valeurs autorisées : 0 à 1024 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	La longueur maximale du journal des commandes Valkey pour les réponses.

Modifications des paramètres de Valkey 8.0

Famille de groupes de paramètres : valkey8

> **Note**
> Redis OSS 7.2.4 est incompatible avec les groupes de paramètres Valkey 8 et supérieurs.

Changements de paramètres spécifiques dans Valkey 8.0

Nom	Détails	Description
repl-backlog-size	Par défaut : 10485760 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	La taille, en octets, de la mémoire tampon du journal des éléments en attente de nœud principal. Le journal des éléments en attente est utilisé pour enregistrer des mises à jour dans les données sur le nœud principal . Lorsqu'une réplique de lecture se connecte au nœud principal, elle tente d'effectuer une synchronisation partielle

Nom	Détails	Description
		(psync), au cours de laquelle elle applique les données du backlog pour rattraper le nœud principal. Si le psync échoue, une synchronisation complète est requise. La valeur minimale pour ce paramètre est 16384. Remarque : à partir de Redis OSS 2.8.22, ce paramètre s'applique au cluster principal ainsi qu'aux répliques de lecture.
nombre maximum d'échantillons de mémoire	Valeur par défaut : 3 Valeurs autorisées : 1 à 64 Type : entier Modifiable : oui Les modifications entrent en vigueur : immédiatement	Pour les calculs least-recently-used (LRU) et time-to-live (TTL), ce paramètre représente la taille de l'échantillon des clés à vérifier. Par défaut, Redis OSS choisit 3 clés et utilise celle qui a été utilisée le moins récemment.

Nouveaux groupes de paramètres dans Valkey 8.0

Nom	Détails	Description
extended-redis-compatibility	Valeurs autorisées : oui, non Par défaut : oui Type : valeur booléenne	Le mode de compatibilité étendue de Redis OSS fait que Valkey se fait passer pour Redis OSS 7.2. Activez cette option uniquement si vous

Nom	Détails	Description

	Modifiable : oui
	
	Quant a lieu la modification : Immédiatement
		rencontrez des problèmes avec les outils ou les clients.
		
		Impacts sur le client :
		
		• `LOADING`- Redis OSS charge le jeu de données en mémoire
		• `BUSY`- Redis OSS est occupé
		• `MISCONF`- Redis OSS est configuré de l'une des manières suivantes :
		  • La `HELLO` commande renvoie « server » => « redis » et « version » => « 7.2.4" (notre version compatible avec Redis OSS).
		  • Le `INFO` champ pour le mode s'appelle « redis_mode ».

Groupes de paramètres supprimés dans Valkey 8.0

Nom	Détails	Description
lazyfree-lazy-eviction	Valeurs autorisées : oui, non Par défaut : non Type : valeur booléenne Modifiable : oui	Effectue une suppression asynchrone des évictions.

Nom	Détails	Description
	Quant a lieu la modification : Immédiatement	
lazyfree-lazy-expire	Valeurs autorisées : oui, non Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Effectue une suppression asynchrone des clés expirées.
lazyfree-lazy-server-del	Valeurs autorisées : oui, non Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Effectue une suppression asynchrone des commandes qui mettent à jour les valeurs.
lazyfree-lazy-user-del	Par défaut : non Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	Lorsque la valeur est définie sur yes, la commande DEL agit de la même manière que UNLINK.

Nom	Détails	Description
replica-lazy-flush	Par défaut : oui Type : valeur booléenne Modifiable : non Ancien nom : slave-lazy-flush	Exécute une commande flushDB asynchrone pendant la synchronisation du réplica.

Modifications des paramètres de Valkey 7.2 et Redis OSS 7

Famille de groupes de paramètres : valkey7

Les groupes de paramètres par défaut de Valkey 7.2 sont les suivants :

- `default.valkey7`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey (mode cluster désactivé) et les groupes de réplication.
- `default.valkey7.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey (mode cluster activé) et les groupes de réplication.

Famille de groupes de paramètres : redis7

Les groupes de paramètres par défaut de Redis OSS 7 sont les suivants :

- `default.redis7`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Redis OSS (mode cluster désactivé) et les groupes de réplication.
- `default.redis7.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Redis OSS (mode cluster activé) et les groupes de réplication.

Modifications de paramètres spécifiques

Les paramètres ajoutés dans Redis OSS 7 sont les suivants. Valkey 7.2 prend également en charge ces paramètres.

Nom	Détails	Description
`cluster-allow-pubsubshard-when-down`	Valeurs autorisées : `yes`, no Par défaut : `yes` Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Lorsque la valeur par défaut est yes, elle autorise les nœuds à traiter le trafic de partition pubsub lorsque le cluster est inactif, à condition qu'il pense être propriétaire des emplacements.
`cluster-preferred-endpoint-type`	Valeurs autorisées : `ip`, `tls-dynamic` Par défaut : `tls-dynamic` Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Cette valeur contrôle le point de terminaison renvoyé pour les requêtes MOVED/ASKING ainsi que le champ de point de terminaison pour `CLUSTER SLOTS` et `CLUSTER SHARDS`. Lorsque la valeur est définie sur ip, le nœud annonce son adresse IP. Lorsque la valeur est définie sur tls-dynamic, le nœud annonce un nom d'hôte lorsqu'il encryption-in-transit est activé et une adresse IP dans le cas contraire.
`latency-tracking`	Valeurs autorisées : `yes`, no Par défaut : no Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	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 distributions de latence cumulées (histogrammes) via la commande `LATENCY`.

Nom	Détails	Description
`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.
`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.

Nom	Détails	Description
`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.

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	Utilisez `listpack` plutôt que `ziplist` pour représenter un petit encodage à hachage

Nom	Détails	Description
	Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	
`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
`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.

Nom	Détails	Description
`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.
`list-max-ziplist-size`	Valeurs autorisées : Par défaut : `-2` Type : entier Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster.	Nombre d'entrées autorisées par nœud de liste interne.

Modifications des paramètres de Redis OSS 6.x

Famille de groupes de paramètres : redis6.x

Les groupes de paramètres par défaut de Redis OSS 6.x sont les suivants :

- `default.redis6.x`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster désactivé) et les groupes de réplication.
- `default.redis6.x.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster activé) et les groupes de réplication.

> **Note**
>
> Dans la version 6.2 du moteur Redis OSS, lorsque la famille de nœuds r6gd a été introduite pour être utilisée avec, seules les politiques de mémoire maximale noeviction Hiérarchisation des données ElastiCache, volatile-lru et allkeys-lru sont prises en charge avec les types de nœuds r6gd.

Pour plus d'informations, consultez ElastiCache version 6.2 pour Redis OSS (améliorée) et ElastiCache version 6.0 pour Redis OSS (améliorée).

Les paramètres ajoutés dans Redis OSS 6.x sont les suivants.

Détails	Description	
`acl-pubsub-default (added in 6.2)`	Valeurs autorisées : `resetchannels` , `allchannels` Par défaut : `allchannels` Type : chaîne Modifiable : oui Les modifications prennent effet : les utilisateurs Redis OSS existants associés au cluster continueront à disposer des autorisations existantes. Mettez à jour les utilisateurs ou redémarrez le cluster pour mettre à jour les utilisateurs Redis OSS existants.	Autorisations de canal Pubsub par défaut pour les utilisateurs ACL déployés sur ce cluster.
`cluster-allow-reads-when-do`	Par défaut : non Type : chaîne	Lorsqu'il est défini sur yes, un groupe de réplication Redis OSS (mode cluster activé) continue de traiter les commandes de lecture

Détails	Description	
`wn (added in 6.0)`	Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	même lorsqu'un nœud n'est pas en mesure d'atteindre un quorum de primaires. Lorsque la valeur par défaut est no, le groupe de réplication rejette toutes les commandes . Nous vous recommandons de définir cette valeur sur yes si vous utilisez un cluster avec moins de trois groupes de nœuds ou si votre application peut gérer des lectures obsolètes en toute sécurité.
`tracking-table-max-keys (added in 6.0)`	Par défaut: 1 000 000 Type : nombre Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	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. Une fois que cette valeur de paramètre est dépassée, les clients reçoivent une invalidation aléatoire. Cette valeur doit être réglée pour limiter l'utilisation de la mémoire tout en gardant une trace de suffisamment de clés. Les clés sont également invalidées dans des conditions de mémoire faible.

Détails	Description	
`acllog-max-len (added in 6.0)`	Valeur par défaut : 128 Type : nombre Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	Cette valeur correspond au nombre maximal d'entrées dans le journal ACL.
`active-expire-effort (added in 6.0)`	Valeur par défaut : 1 Type : nombre Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	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.

Détails	Description	
`lazyfree-lazy-user-del (added in 6.0)`	Par défaut : non Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	Lorsque la valeur est définie sur yes, la propriété `DEL` agit de la même manière que `UNLINK`.

Les paramètres supprimés dans Redis OSS 6.x sont les suivants.

Nom	Détails	Description
`lua-replicate-commands`	Valeurs autorisées : yes / no Par défaut : oui Type : valeur booléenne Modifiable : oui Les modificat ions prennent effet : immédiate ment	Active ou non la réplication de l'effet Lua dans les scripts Lua

Modifications des paramètres de Redis OSS 5.0.3

Famille de groupes de paramètres : redis5.0

Groupes de paramètres par défaut de Redis OSS 5.0

- `default.redis5.0`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster désactivé) et les groupes de réplication.

- `default.redis5.0.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster activé) et les groupes de réplication.

Paramètres ajoutés dans Redis OSS 5.0.3

Nom	Détails	Description
`rename-commands`	Par défaut: Aucun Type : chaîne Modifiable : oui Les modifications prennent effet : immédiatement sur tous les nœuds du cluster	Liste séparée par des espaces de commandes Redis OSS renommées. Voici une liste limitée de commandes disponibles pour le changement de nom : `APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLPUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFLOAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPUSH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLPUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET`

Nom	Détails	Description
		`SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP SRANDMEMBER SREM STRLEN SUBSCRIBE SUNION SUNIONSTORE SWAPDB TIME TOUCH TTL TYPE UNSUBSCRIBE UNLINK UNWATCH WAIT WATCH ZADD ZCARD ZCOUNT ZINCRBY ZINTERSTORE ZLEXCOUNT ZPOPMAX ZPOPMIN ZRANGE ZRANGEBYLEX ZREVRANGEBYLEX ZRANGEBYSCORE ZRANK ZREM ZREMRANGEBYLEX ZREMRANGEBYRANK ZREMRANGEBYSCORE ZREVRANGE ZREVRANGEBYSCORE ZREVRANK ZSCORE ZUNIONSTORE SCAN SSCAN HSCAN ZSCAN XINFO XADD XTRIM XDEL XRANGE XREVRANGE XLEN XREAD XGROUP XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_RO LOLWUT XSETID SUBSTR`

Pour de plus amples informations, veuillez consulter ElastiCache version 5.0.6 pour Redis OSS (améliorée).

Modifications des paramètres de Redis OSS 5.0.0

Famille de groupes de paramètres : redis5.0

Groupes de paramètres par défaut de Redis OSS 5.0

- `default.redis5.0`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster désactivé) et les groupes de réplication.
- `default.redis5.0.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster activé) et les groupes de réplication.

Paramètres ajoutés dans Redis OSS 5.0

Nom	Détails	Description
`stream-node-max-bytes`	Valeurs autorisées : 0+ Par défaut: 4096 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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`	Valeurs autorisées : 0+ Par défaut : 100 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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. Si la valeur est définie sur 0, le nombre d'éléments du nœud de l'arborescence n'est pas limité.
`active-defrag-max-scan-fields`	Valeurs autorisées : de 1 à 1000000 Par défaut: 1000 Type : entier Modifiable : oui	Nombre maximum de set/hash/zset/list champs qui seront traités à partir de l'analyse du dictionnaire principal

Nom	Détails	Description
	Les modificat ions prennent effet : immédiate ment	
`lua-replicate-commands`	Valeurs autorisées : yes / no Par défaut : oui Type : valeur booléenne Modifiable : oui Les modificat ions prennent effet : immédiate ment	Active ou non la réplication de l'effet Lua dans les scripts Lua
`replica-ignore-maxmemory`	Par défaut : oui Type : valeur booléenne Modifiable : non	Détermine si le réplica ignore le paramètre `maxmemory` en ne supprimant pas les éléments indépendants du principal

Redis OSS a renommé plusieurs paramètres dans la version 5.0 du moteur en réponse aux commentaires de la communauté. Pour plus d'informations, consultez Quoi de neuf dans Redis OSS 5 ? . Le tableau suivant répertorie les nouveaux noms et montre leur correspondance avec les versions précédentes.

Paramètres renommés dans Redis OSS 5.0

Nom	Détails	Description
`replica-lazy-flush`	Par défaut : oui Type : valeur booléenne Modifiable : non	Exécute une commande flushDB asynchrone pendant la synchronisation du réplica.

Nom	Détails	Description
	Ancien nom : slave-lazy-flush	
`client-output-buffer-limit-replica-hard-limit`	Par défaut : pour connaître les valeurs, consultez Paramètres spécifiques au type de nœud Redis OSS Type : entier Modifiable : non Ancien nom : client-output-buffer-limit - slave-hard-limit	Pour les répliques de lecture Redis OSS : si le tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté.
`client-output-buffer-limit-replica-soft-limit`	Par défaut : pour connaître les valeurs, consultez Paramètres spécifiques au type de nœud Redis OSS Type : entier Modifiable : non Ancien nom : client-output-buffer-limit - slave-soft-limit	Pour les répliques de lecture 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-replica-soft-seconds`
`client-output-buffer-limit-replica-soft-seconds`	Par défaut : 60 Type : entier Modifiable : non Ancien nom : client-output-buffer-limit - slave-soft-seconds	Pour les répliques de lecture Redis OSS : si la mémoire tampon de sortie d'un client reste en `client-output-buffer-limit-replica-soft-limit` octets pendant plus de secondes, le client sera déconnecté.

Nom	Détails	Description
`replica-allow-chaining`	Par défaut : non Type : chaîne Modifiable : non Ancien nom : slave-allow-chaining	Détermine si une réplique en lecture dans Redis OSS peut avoir ses propres répliques en lecture.
`min-replicas-to-write`	Par défaut : 0 Type : entier Modifiable : oui Ancien nom : min-slaves-to-write Les modificat ions prennent effet : immédiate ment	Le nombre minimum de réplicas en lecture qui doivent être disponibles pour que le nœud principal puisse accepter des écritures des clients. Si le nombre de réplicas disponibles est inférieur à ce nombre, alors le nœud principal n'acceptera plus de demandes d'écriture. Si ce paramètre min-replicas-max-lag est égal à 0, le nœud principal acceptera toujours les demandes d'écriture, même si aucune réplique n'est disponible.
`min-replicas-max-lag`	Par défaut: 10 Type : entier Modifiable : oui Ancien nom : min-slaves-max-lag Les modificat ions prennent effet : immédiate ment	Le nombre de secondes au cours desquelles le nœud principal doit suivre une requête ping à partir d'un réplica en lecture. Si ce laps de temps s'écoule et que le nœud principal ne reçoit pas un message ping, alors le réplica n'est plus considéré comme étant disponible. Si le nombre de répliques disponibles tombe en dessous min-replicas-to-write, le serveur principal cessera d'accepter les écritures à ce moment-là. Si ce paramètre min-replicas-to-write est égal à 0, le nœud principal acceptera toujours les demandes d'écriture, même si aucune réplique n'est disponible.

Nom	Détails	Description
`close-on-replica-write`	Par défaut : oui Type : valeur booléenne Modifiable : oui Ancien nom : close-on-slave-write Les modificat ions prennent effet : immédiate ment	Si activé, les clients qui tentent d'écrire sur un réplica en lecture seule seront déconnectés.

Paramètres supprimés dans Redis OSS 5.0

Nom	Détails	Description
`repl-timeout`	Par défaut : 60 Modifiable : non	Ce paramètre n'est pas disponible dans cette version.

Modifications des paramètres de Redis OSS 4.0.10

Famille de groupes de paramètres : redis4.0

Groupes de paramètres par défaut de Redis OSS 4.0.x

- `default.redis4.0`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster désactivé) et les groupes de réplication.
- `default.redis4.0.cluster.on`— Utilisez ce groupe de paramètres, ou un groupe dérivé de celui-ci, pour les clusters Valkey ou Redis OSS (mode cluster activé) et les groupes de réplication.

Paramètres modifiés dans Redis OSS 4.0.10

Nom	Détails	Description
`maxmemory-policy`	Valeurs autorisées : `allkeys-lru` , `volatile-lru` , **`allkeys-lfu`** , **`volatile-lfu`** , `allkeys-random` , `volatile-random` , `volatile-ttl` , `noeviction` Par défaut : volatile-lru Type : chaîne Modifiable : oui Quant a lieu la modification : Immédiatement	La rubrique `maxmemory-policy` a été ajoutée dans la version 2.6.13. Dans la version 4.0.10, deux nouvelles valeurs autorisées ont été ajoutées : `allkeys-lfu` (qui élimine toute clé utilisant une LFU approximative) et `volatile-lfu` (qui élimine l'utilisation de LFU approximatives parmi les clés avec un ensemble d'expiration). Dans la version 6.2, lorsque la famille de nœuds r6gd a été introduit e pour être utilisée avec la hiérarchisation des données, seules les politiques de mémoire maximale `noeviction` , `volatile-lru` et `allkeys-lru` sont prises en charge avec les types de nœuds r6gd.

Paramètres ajoutés dans Redis OSS 4.0.10

Nom	Détails	Description
Paramètres de suppression asynchrones		
`lazyfree-lazy-eviction`	Valeurs autorisées : yes / no Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Effectue une suppression asynchrone des évictions.

Nom	Détails	Description
`lazyfree-lazy-expire`	Valeurs autorisées : yes / no Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Effectue une suppression asynchrone des clés expirées.
`lazyfree-lazy-server-del`	Valeurs autorisées : yes / no Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Effectue une suppression asynchrone des commandes qui mettent à jour les valeurs.
`slave-lazy-flush`	Valeurs autorisées : N/A Par défaut : non Type : valeur booléenne Modifiable : non Quant a lieu la modification : N/A	Exécute une commande flushDB asynchrone pendant la synchronisation esclave.
Paramètres d'LFU		

Nom	Détails	Description
`lfu-log-factor`	Valeurs autorisées : tous les entiers > 0 Par défaut: 10 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Définissez le facteur journal qui détermine le nombre d'accès à une clé pour en saturer le compteur.
`lfu-decay-time`	Valeurs autorisées : Tous les entiers Valeur par défaut : 1 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Le temps nécessaire en minutes pour réduire le compteur de clé.
Paramètres de défragmentation active		
`activedefrag`	Valeurs autorisées : yes / no Par défaut : non Type : valeur booléenne Modifiable : oui Quant a lieu la modification : Immédiatement	Activation de la défragmentation active.

Nom	Détails	Description
`active-defrag-ignore-bytes`	Valeurs autorisées : 10485760-104857600 Par défaut: 104857600 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Quantité minimum de fragmentation perdue pour lancer une défragmentation active.
`active-defrag-threshold-lower`	Valeurs autorisées : 1-100 Par défaut: 10 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Pourcentage minimum de fragmentation pour lancer une défragmentation active.
`active-defrag-threshold-upper`	Valeurs autorisées : 1-100 Par défaut : 100 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Pourcentage maximum de fragmentation à partir duquel nous utilisons l'effort maximum.

Nom	Détails	Description
`active-defrag-cycle-min`	Valeurs autorisées : 1-75 Par défaut: 25 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Effort minimum pour défragmenter en pourcentage d'UC.
`active-defrag-cycle-max`	Valeurs autorisées : 1-75 Par défaut: 75 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Effort maximum pour défragmenter en pourcentage d'UC.
Paramètres du tampon de sortie client		
`client-query-buffer-limit`	Valeurs autorisées : 1048576-1073741824 Par défaut: 1073741824 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Taille maximum d'un seul tampon de requête client.

Nom	Détails	Description
`proto-max-bulk-len`	Valeurs autorisées : 1048576-536870912 Par défaut: 536870912 Type : entier Modifiable : oui Quant a lieu la modification : Immédiatement	Taille maximum d'une seule demande d'élément.

Modifications des paramètres de Redis OSS 3.2.10

Famille de groupes de paramètres : redis3.2

ElastiCache pour Redis OSS 3.2.10, aucun paramètre supplémentaire n'est pris en charge.

Modifications des paramètres de Redis OSS 3.2.6

Famille de groupes de paramètres : redis3.2

Pour Redis OSS 3.2.6, aucun paramètre supplémentaire n'est pris en charge.

Modifications des paramètres de Redis OSS 3.2.4

Famille de groupes de paramètres : redis3.2

À partir de Redis OSS 3.2.4, il existe deux groupes de paramètres par défaut.

- `default.redis3.2`— Lorsque vous exécutez Redis OSS 3.2.4, spécifiez ce groupe de paramètres ou un groupe dérivé de celui-ci, si vous souhaitez créer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) tout en utilisant les fonctionnalités supplémentaires de Redis OSS 3.2.4.
- `default.redis3.2.cluster.on`— Spécifiez ce groupe de paramètres ou un groupe dérivé de celui-ci lorsque vous souhaitez créer un groupe de réplication Valkey ou Redis OSS (mode cluster activé).

Rubriques

- Nouveaux paramètres pour Redis OSS 3.2.4
- Paramètres modifiés dans Redis OSS 3.2.4 (amélioré)

Nouveaux paramètres pour Redis OSS 3.2.4

Famille de groupes de paramètres : redis3.2

Pour Redis OSS 3.2.4, les paramètres supplémentaires suivants sont pris en charge.

Nom	Détails	Description
`list-max-ziplist-size`	Par défaut : -2 Type : entier Modifiable : non	Les listes sont codées de manière spécifiqu e pour économiser de l'espace. Le nombre d'entrées autorisées par le nœud de liste interne peut être spécifié comme une taille maximale fixe ou sous forme d'un nombre maximal d'éléments. Pour une taille maximale fixe, utilisez -5 à -1, ce qui signifie : • -5 : taille maximale : 64 Ko - non recommand é pour les charges de travail habituelles • -4 : taille maximale : 32 Ko - non recommand é • -3 : taille maximale : 16 Ko - non recommand é • -2 : taille maximale : 8 Ko - conseillé • -1 : taille maximale : 4 Ko - conseillé • Les nombres positifs signifient un stockage jusqu'au nombre exact d'éléments par nœud de liste.

Nom	Détails	Description
`list-compress-depth`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Les listes peuvent également être compressé es. 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 opération s 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. • Etc.

Nom	Détails	Description
`cluster-enabled`	Par défaut : no/yes * Type : chaîne Modifiable : non	Indique s'il s'agit d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) en mode cluster (oui) ou d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) en mode non-cluster (non). Les groupes de réplication Valkey ou Redis OSS (mode cluster activé) en mode cluster peuvent partitionner leurs données sur un maximum de 500 groupes de nœuds. * Redis OSS 3.2. x possède deux groupes de paramètres par défaut. • `default.redis3.2` – valeur par défaut no. • `default.redis3.2.cluster.on` – valeur par défaut `yes`. .

Nom	Détails	Description
`cluster-require-full-coverage`	Par défaut : non Type : valeur booléenne Modifiable : oui Les modificat ions prennent effet : immédiate ment	Lorsqu'ils sont définis sur`yes`, les nœuds Valkey ou Redis OSS (mode cluster activé) en mode cluster cessent d'accepter les requêtes s'ils détectent qu'au moins un emplacement de hachage est découvert (aucun nœud disponibl e ne le dessert). De cette façon, si le cluster est partiellement défaillant, le cluster n'est pas disponible. Il redevient automatiquement disponible dès que tous les emplacements sont couverts à nouveau. Cependant, vous souhaitez parfois que le sous-ensemble du cluster qui fonctionne continue à accepter des requêtes pour la partie de l'espace de clé qui est toujours couverte. Pour cela, il suffit de définir l'option `cluster-r` `equire-full-coverage` sur no.

Nom	Détails	Description
`hll-sparse-max-bytes`	Par défaut: 3000 Type : entier Modifiable : oui Les modifications prennent effet : immédiatement	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 3 000 afin de bénéficier des avantages du codage efficace en espace sans trop ralentir PFADD, qui est O(N) avec le codage fragmenté. 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.

Nom	Détails	Description
`reserved-memory-percent`	Par défaut: 25 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Pourcentage de mémoire d'un nœud réservé à des fins autres que le traitement des données. Par défaut, l'empreinte de données Redis OSS augmente jusqu'à ce qu'elle consomme toute la mémoire du nœud. Si cela se produit, les performances du nœud seront probablem ent affectées en raison d'une pagination de mémoire excessive. En réservant de la mémoire, vous pouvez réserver une partie de la mémoire disponible à des fins autres que REDIS OSS afin de réduire la quantité de pagination. Ce paramètre est spécifique à la ElastiCache distribution standard Redis OSS et ne fait pas partie de celle-ci. Pour plus d'informations, consultez `reserved-memory` et Gestion de la mémoire réservée pour Valkey et Redis OSS.

Paramètres modifiés dans Redis OSS 3.2.4 (amélioré)

Famille de groupes de paramètres : redis3.2

Pour Redis OSS 3.2.4, les paramètres suivants ont été modifiés.

Nom	Détails	Modification
`activerehashing`	Modifiable : oui si le groupe de paramètres n'est associé à aucun cluster de cache. Dans le cas contraire, ce champ n'est pas obligatoire.	Modifiable était défini sur Non.

Nom	Détails	Modification
`databases`	Modifiable : oui si le groupe de paramètres n'est associé à aucun cluster de cache. Dans le cas contraire, ce champ n'est pas obligatoire.	Modifiable était défini sur Non.
`appendonly`	Valeur par défaut : désactivé Modifiable : non	Si vous souhaitez effectuer une mise à niveau depuis une version antérieure de Redis OSS, vous devez d'abord la `appendonly` désactiver.
`appendfsync`	Valeur par défaut : désactivé Modifiable : non	Si vous souhaitez effectuer une mise à niveau depuis une version antérieure de Redis OSS, vous devez d'abord la `appendfsync` désactiver.
`repl-timeout`	Par défaut : 60 Modifiable : non	Est désormais non modifiable avec une valeur par défaut de 60.
`tcp-keepalive`	Valeur par défaut : 300	La valeur par défaut était 0.
`list-max-ziplist-entries`		Le paramètre n'est plus disponible.
`list-max-ziplist-value`		Le paramètre n'est plus disponible.

Paramètres ajoutés à Redis OSS 2.8.24 (amélioré)

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.24, aucun paramètre supplémentaire n'est pris en charge.

Paramètres ajoutés à Redis OSS 2.8.23 (amélioré)

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.23, le paramètre supplémentaire suivant est pris en charge.

Nom	Détails	Description
`close-on-slave-write`	Par défaut : oui Type : chaîne (oui/non) Modifiable : oui Les modificat ions prennent effet : immédiate ment	Si activé, les clients qui tentent d'écrire sur un réplica en lecture seule seront déconnectés.

Fonctionnement d' close-on-slave-write

`close-on-slave-write`Ce paramètre a été introduit par Amazon ElastiCache pour vous permettre de mieux contrôler la façon dont votre cluster réagit lorsqu'un nœud principal et un nœud de réplique de lecture échangent des rôles en raison de la promotion d'une réplique de lecture vers un nœud principal.

Si le cluster réplica en lecture devient cluster réplica principal pour une raison autre qu'un basculement du groupe de réplication avec Multi-AZ activé, le client continuera d'essayer d'écrire

sur le point de terminaison A. Etant donné que le point de terminaison A est désormais le point de terminaison d'un réplica en lecture, ces écritures échoueront. C'est le comportement de Redis OSS avant son ElastiCache introduction `close-on-replica-write` et le comportement si vous le désactivez`close-on-replica-write`.

Avec `close-on-replica-write` activé, à chaque fois qu'un client tente d'écrire dans un réplica en lecture, la connexion client au cluster prend fin. Votre logique d'application doit détecter la déconnexion, vérifiez la table DNS et reconnectez-vous au point de terminaison principal, qui sera désormais le point de terminaison B.

Quand vous pouvez désactiver close-on-replica-write

Si la désactivation du paramètre `close-on-replica-write` génère des écritures dans le cluster en échec, pourquoi désactiver le paramètre `close-on-replica-write` ?

Comme mentionné précédemment, lorsque ce paramètre `close-on-replica-write` est activé, dès qu'un client tente d'écrire dans un réplica en lecture, la connexion client au cluster s'interrompt. Établir une nouvelle connexion avec le nœud prend du temps. Par conséquent, déconnecter et reconnecter suite à une demande d'écriture sur le réplica affecte également la latence des demandes

de lecture diffusées par le biais de la même connexion. Cet effet reste actif jusqu'à ce que la nouvelle connexion soit établie. Si votre application a une charge particulièrement lourde en écriture ou si elle est très sensible au temps de latence, vous souhaiterez peut-être que vos clients restent connectés afin de ne pas dégrader les performances de lecture.

Paramètres ajoutés à Redis OSS 2.8.22 (amélioré)

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.22, aucun paramètre supplémentaire n'est pris en charge.

> **Important**
>
> - À partir de la version 2.8.22 de Redis OSS, `repl-backlog-size` elle s'applique au cluster principal ainsi qu'aux clusters de réplication.
> - À partir de la version 2.8.22 de Redis OSS, le `repl-timeout` paramètre n'est pas pris en charge. S'il est modifié, il ElastiCache sera remplacé par la valeur par défaut (60s), comme nous le faisons avec`appendonly`.

Les paramètres suivants ne sont plus pris en charge.

- appendonly
- appendfsync
- repl-timeout

Paramètres ajoutés à Redis OSS 2.8.21

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.21, aucun paramètre supplémentaire n'est pris en charge.

Paramètres ajoutés à Redis OSS 2.8.19

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.19, aucun paramètre supplémentaire n'est pris en charge.

Paramètres ajoutés à Redis OSS 2.8.6

Famille de groupes de paramètres : redis2.8

Pour Redis OSS 2.8.6, les paramètres supplémentaires suivants sont pris en charge.

Nom	Détails	Description
`min-slaves-max-lag`	Par défaut: 10 Type : entier Modifiable : oui Les modificat ions prennent effet : imn ment	Le nombre de secondes au cours desquelles le nœud principal doit suivre une requête ping à partir d'un réplica en lecture. Si ce laps de temps s'écoule et que le nœud principal ne reçoit pas un message ping, alors le réplica n'est plus considéré comme étant disponible. Si le nombre de répliques disponibl es tombe en dessous min-slave s-to-write, le serveur principal cessera d'accepter les écritures à ce moment-là. Si ce paramètre min-slaves- to-write est égal à 0, le nœud principal acceptera toujours les demandes d'écriture, même si aucune réplique n'est disponible.
`min-slaves-to-write`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : imn ment	Le nombre minimum de réplicas en lecture qui doivent être disponibles pour que le nœud principal puisse accepter des écritures des clients. Si le nombre de réplicas disponibles est inférieur à ce nombre, alors le nœud principal n'acceptera plus de demandes d'écriture. Si ce paramètre min-slaves- max-lag est égal à 0, le nœud principal acceptera toujours les

Nom	Détails	Description
		demandes d'écriture, même si aucune réplique n'est disponible.

Nom	Détails	Description
`notify-keyspace-events`	Par défaut : (une chaîne vide) Type : chaîne Modifiable : oui Les modificat ions prennent effet : imm ment	Les types d'événements keyspace dont Redis OSS peut informer les clients. Chaque type d'événeme nt est représenté par une lettre unique : • K — Événements de Keyspaces publiés avec le préfixe __keyspace@<db>__ • E — Événements d'événeme nt-clé, publiés avec le préfixe __keyevent@<db>__ • g — Commandes génériques, non spécifiques, telles que DEL, EXPIRE, RENAME, etc. • $ – Commandes de texte • l — Commandes de liste • s — Commandes d'ensemble • h — Commandes de hachage • z — Commandes d'ensemble trié • x — Événements arrivés à expiration (événements générés chaque fois qu'une clé arrive à expiration) •

Nom	Détails	Description
		e — Événements expulsés (événements générés quand une clé est expulsée pour des raisons de mémoire maximum atteinte) • A — Alias pour g$lshzxe Vous pouvez avoir une combinais on de ces types d'événements. Par exemple, AKE signifie que Redis OSS peut publier des notifications de tous les types d'événements. N'utilisez pas de caractères autres que ceux mentionnés dans la liste ci-dessus, car cela produirait des messages d'erreur. Par défaut, ce paramètre est défini sur une chaîne vide, ce qui signifie que la notification d'événement d'espace de clé est désactivée.

Nom	Détails	Description
`repl-backlog-size`	Par défaut: 1048576 Type : entier Modifiable : oui Les modificat ions prennent effet : imn ment	La taille, en octets, de la mémoire tampon du journal des éléments en attente de nœud principal. Le journal des éléments en attente est utilisé pour enregistrer des mises à jour dans les données sur le nœud principal. Lorsqu'un réplica en lecture se connecte au réplica principal, il tente d'effectuer une synchronisation partielle (`psync`), où il applique les données du journal des éléments en attente avec le nœud principal. Si le `psync` échoue, une synchroni sation complète doit être effectuée. La valeur minimale pour ce paramètre est 16384. **Note** À partir de Redis OSS 2.8.22, ce paramètre s'applique au cluster principal ainsi qu'aux répliques de lecture.

Nom	Détails	Description
`repl-backlog-ttl`	Par défaut: 3600 Type : entier Modifiable : oui Les modificat ions prennent effet : imn ment	Le nombre de secondes pendant lesquelles le nœud principal conserve la mémoire tampon du journal des éléments en attente. Dès que le dernier nœud réplica est déconnecté, les données du journal des éléments en attente sont conservées jusqu'à l'expirat ion du paramètre `repl-back log-ttl` . Si le réplica n'est pas connecté au réplica principal au cours de cette période, alors le réplica principal libère la mémoire tampon du journal des éléments en attente. Lorsque le réplica se reconnecte, il devra effectuer une synchronisation complète avec le réplica principal. Si ce paramètre est défini sur 0, alors la mémoire tampon du journal des éléments en attente ne sera jamais libérée.

Nom	Détails	Description
`repl-timeout`	Par défaut : 60 Type : entier Modifiable : oui Les modificat ions prennent effet : imm ment	Représente le délai d'expiration, en secondes, pour : • le transfert de données en bloc lors de la synchronisation, du côté du réplica en lecture ; • l'expiration du nœud principal du côté du réplica ; • l'expiration du réplica du côté du nœud principal.

Paramètres de Redis OSS 2.6.13

Famille de groupes de paramètres : redis2.6

Redis OSS 2.6.13 était la première version de Redis OSS prise en charge par. ElastiCache Le tableau suivant indique les paramètres pris en charge par Redis OSS 2.6.13. ElastiCache

Nom	Détails	Description
`activereh ashing`	Par défaut : oui Type : chaîne (oui/non) Modifiable : oui Quand a lieu la modification : à la création	Détermine s'il faut activer la nouvelle fonction de hachage active de Redis. 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.

Nom	Détails	Description
`appendonly`	Par défaut : non Type : chaîne Modifiable : oui Les modificat ions prennent effet : immédiate ment	Active ou désactive la fonctionnalité AOF (Append-Only File) de Redis. L'AOF capture toutes les commandes Redis OSS qui modifient les données du cache et est utilisée pour effectuer une restauration après une défaillance de certains nœuds. La valeur par défaut est non, ce qui signifie que la fonctionnalité AOF est désactivée. Définissez ce paramètre sur oui pour activer AOF. Pour de plus amples informations, veuillez consulter Atténuation des défaillances. **Note** Les fichiers AOF ne sont pas pris en charge pour les nœuds cache.t1.micro et cache.t2.*. Pour les nœuds de ce type, la valeur du paramètre `appendonly` est ignorée. **Note** Pour les groupes de réplication Multi-AZ, AOF n'est pas autorisée.

Nom	Détails	Description
`appendfsync`	Par défaut : everysec Type : chaîne Modifiable : oui Les modificat ions prennent effet : immédiate ment	Lorsque `appendonly` est défini sur yes, contrôle la fréquence à laquelle le tampon de sortie AOF est écrit sur le disque : • non – le tampon est vidé sur le disque en fonction des besoins. • everysec — le tampon est vidé chaque seconde. Il s'agit de l'option par défaut. • always — le tampon est vidé chaque fois que les données du cluster sont modifiées. • Appendfsync n'est pas pris en charge pour les versions 2.8.22 et ultérieures.
`client-output-buffer-limit-normal-hard-limit`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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).

Nom	Détails	Description
`client-output-buffer-limit-normal-soft-seconds`	Par défaut : 0 Type : entier Modifiable : oui Les modifications prennent effet : immédiatement	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).
`client-output-buffer-limit-pubsub-hard-limit`	Par défaut: 33554432 Type : entier Modifiable : oui Les modifications prennent effet : immédiatement	Pour les publish/subscribe clients 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`	Par défaut: 8388608 Type : entier Modifiable : oui Les modifications prennent effet : immédiatement	Pour les publish/subscribe clients 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`	Par défaut : 60 Type : entier Modifiable : oui Les modifications prennent effet : immédiatement	Pour les publish/subscribe clients 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
`client-output-buffer-limit-slave-hard-limit`	Par défaut : pour connaître les valeurs, consultez Paramètres spécifiques au type de nœud Redis OSS Type : entier Modifiable : non	Pour les répliques de lecture Redis OSS : si le tampon de sortie d'un client atteint le nombre d'octets spécifié, le client sera déconnecté.
`client-output-buffer-limit-slave-soft-limit`	Par défaut : pour connaître les valeurs, consultez Paramètres spécifiques au type de nœud Redis OSS Type : entier Modifiable : non	Pour les répliques de lecture 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-slave-soft-seconds`
`client-output-buffer-limit-slave-soft-seconds`	Par défaut : 60 Type : entier Modifiable : non	Pour les répliques de lecture Redis OSS : si la mémoire tampon de sortie d'un client reste en `client-output-buffer-limit-slave-soft-limit` octets pendant plus de secondes, le client sera déconnecté.
`databases`	Par défaut: 16 Type : entier Modifiable : non Quand a lieu la modification : à la création	Le nombre de partitions logiques des bases de données. Nous recommandons de conserver cette valeur basse. 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.

Nom	Détails	Description
`hash-max-ziplist-entries`	Par défaut: 512 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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`	Par défaut: 64 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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.
`list-max-ziplist-entries`	Par défaut: 512 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Détermine la quantité de mémoire utilisée pour les listes. Les listes avec un nombre d'entrées inférieur à celui spécifié sont stockées à l'aide d'un encodage spécial qui économise de l'espace.
`list-max-ziplist-value`	Par défaut: 64 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Détermine la quantité de mémoire utilisée pour les listes. Les listes avec des entrées qui ont un nombre d'octets plus petit que le nombre spécifié sont stockées à l'aide d'un encodage spécial qui économise de l'espace.

Nom	Détails	Description
`lua-time-limit`	Par défaut : 5000 Type : entier Modifiable : non	Durée maximale d'exécution d'un script Lua, en millisecondes, avant qu'une action ne soit entreprise pour ElastiCache arrêter le script. En cas `lua-time-limit` de dépassement, toutes les commandes Redis OSS renverront une erreur de la forme ____-BUSY. Étant donné que cet état peut interférer avec de nombreuses opérations essentielles de Redis OSS, je ElastiCache vais d'abord émettre une commande SCRIPT KILL. En cas d'échec, Redis ElastiCache OSS sera redémarré de force.
`maxclients` Cette valeur s'applique à tous les types d'instance, à l'exception de ceux spécifiés explicitement	Par défaut: 65000 Type : entier Modifiable : non	Le nombre maximum de clients qui peut être connecté à un moment donné.
	t2.medium par défaut : 20000 Type : entier Modifiable : non	
	Par défaut t2.small : 20000 Type : entier Modifiable : non	
	Par défaut t2.micro : 20000 Type : entier Modifiable : non	

Nom	Détails	Description
	Par défaut t4g.micro : 20000 Type : entier Modifiable : non	
	Pt3.medium par défaut : 46000 Type : entier Modifiable : non	
	Par défaut t3.small : 46000 Type : entier Modifiable : non	
	Par défaut t3.micro : 20000 Type : entier Modifiable : non	
`maxmemory-policy`	Par défaut : volatile-lru Type : chaîne Modifiable : oui Les modificat ions prennent effet : immédiate ment	La politique d'expulsion des clés lors de l'utilisa tion de la mémoire maximale est atteinte. Les valeurs valides sont : `volatile-lru | allkeys-lru | volatile-random | allkeys-random | volatile-ttl | noeviction` Pour plus d'informations, consultez la section Utilisation de Valkey ou Redis OSS comme cache LRU.

Nom	Détails	Description
`maxmemory-samples`	Valeur par défaut : 3 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Pour les calculs least-recently-used (LRU) et time-to-live (TTL), ce paramètre représente la taille de l'échantillon des clés à vérifier. Par défaut, Redis OSS choisit 3 clés et utilise celle qui a été utilisée le moins récemment.
`reserved-memory`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Mémoire totale réservée, en octets, pour une utilisation autre que le traitement des données. Par défaut, le nœud Redis OSS s'agrandi ra jusqu'à ce qu'il consomme celui du nœud `maxmemory` (voirParamètres spécifiques au type de nœud Redis OSS). Si cela se produit, les performances du nœud seront probablem ent affectées en raison d'une pagination de mémoire excessive. En réservant de la mémoire, vous pouvez réserver une partie de la mémoire disponible à des fins autres que REDIS OSS afin de réduire la quantité de pagination. Ce paramètre est spécifique à la ElastiCache distribution standard Redis OSS et ne fait pas partie de celle-ci. Pour plus d'informations, consultez `reserved-memory-percent` et Gestion de la mémoire réservée pour Valkey et Redis OSS.

Nom	Détails	Description
`set-max-intset-entries`	Par défaut: 512 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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.
`slave-allow-chaining`	Par défaut : non Type : chaîne Modifiable : non	Détermine si une réplique en lecture dans Redis OSS peut avoir ses propres répliques en lecture.
`slowlog-log-slower-than`	Par défaut: 10000 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Durée d'exécution maximale, en microseco ndes, des commandes à enregistrer par la fonction Redis OSS Slow Log.
`slowlog-max-len`	Valeur par défaut : 128 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	La longueur maximale du journal lent de Redis OSS.

Nom	Détails	Description
`tcp-keepalive`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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. **Important** Certains aspects de ce paramètre ont changé dans la version 3.2.4 de Redis OSS. Consultez Paramètres modifiés dans Redis OSS 3.2.4 (amélioré).
`timeout`	Par défaut : 0 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	Le nombre de secondes qu'un nœud doit attendre avant d'être mis hors service. Les valeurs sont les suivantes : • `0` – ne jamais déconnecter un client inactif. • `1-19` – valeurs non valides. • `>=20` – le nombre de secondes qu'un nœud doit attendre avant de déconnecter un client inactif.
`zset-max-ziplist-entries`	Valeur par défaut : 128 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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.

Nom	Détails	Description
`zset-max-ziplist-value`	Par défaut: 64 Type : entier Modifiable : oui Les modificat ions prennent effet : immédiate ment	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.

> **Note**
>
> Si vous ne spécifiez aucun groupe de paramètres pour votre cluster Redis OSS 2.6.13, un groupe de paramètres par défaut (`default.redis2.6`) sera utilisé. Vous ne pouvez pas modifier les valeurs des paramètres dans un groupe de paramètres par défaut ; cependant, vous pouvez créer un groupe de paramètres personnalisé et l'affecter à votre cluster à tout moment.

Paramètres spécifiques au type de nœud Redis OSS

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 les valeurs par défaut des paramètres `maxmemory`, `client-output-buffer-limit-slave-hard-limit` et `client-output-buffer-limit-slave-soft-limit` 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. Pour de plus amples informations, veuillez consulter Mémoire disponibe.

> **Note**
>
> Le paramètre `maxmemory` ne peut pas être modifié.

Type de nœud	Maxmemory	C lient-output-buffe r-limit - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g. petit	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880
cache.m2.2xlarge	35022438400	3502243840	3502243840
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120

Type de nœud	Maxmemory	C lient-output-buffe r-limit - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172
cache.m6g.2xlarge	27966669210	2796666921	2796666921
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199

Type de nœud	Maxmemory	C lient-output-buffe r-limit - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3.2xlarge	62495129600	6081740800	6081740800
cache.r3.4xlarge	126458265600	12268339200	12268339200
cache.r3.8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970
cache.r5.2xlarge	56711183565	5671118356	5671118356
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381

Type de nœud	Maxmemory	C lient-output-buffe r-limit - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970
cache.r7g.2xlarge	56711183565	5671118356	5671118356
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655

Type de nœud	Maxmemory	C lient-output-buffe r-limit - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

> **Note**
>
> Tous les types d'instances de la génération actuelle sont créés dans Amazon Virtual Private Cloud (VPC) par défaut.
> Les instances T1 ne prennent pas en charge Multi-AZ.
> Les instances T1 et T2 ne prennent pas en charge Redis OSS AOF.

Les variables de configuration Redis OSS ne `appendfsync` sont pas prises en charge sur Redis OSS version 2.8.22 `appendonly` et versions ultérieures.

Paramètres spécifiques Memcached

Memcached

Si vous ne spécifiez pas un groupe de paramètres pour votre cluster Memcached, alors un groupe de paramètres approprié pour votre version de moteur sera utilisé. Vous ne pouvez pas modifier les valeurs des paramètres dans un groupe de paramètres par défaut. Cependant, vous pouvez créer un groupe de paramètres personnalisés et l'assigner à votre cluster à tout moment. Pour de plus amples informations, veuillez consulter Création d'un groupe ElastiCache de paramètres.

Rubriques

- Changements apportés à Memcached 1.6.17
- Paramètres ajoutés Memcached 1.6.6
- Modifications des paramètres Memcached 1.5.10
- Paramètres ajoutés Memcached 1.4.34
- Paramètres ajoutés Memcached 1.4.33
- Paramètres ajoutés Memcached 1.4.24
- Paramètres ajoutés Memcached 1.4.14
- Paramètres Memcached 1.4.5 pris en charge
- Surcharge de la connexion Memcached
- Paramètres propres au type de nœud Memcached

Changements apportés à Memcached 1.6.17

À partir de Memcached 1.6.17, les commandes administratives `lru_crawler`, `lru` et `slabs` ne sont plus prises en charge. Avec ces modifications, vous ne pourrez pas le faire au enable/disable `lru_crawler` moment de l'exécution via des commandes. Veuillez enable/disable `lru_crawler` modifier votre groupe de paramètres personnalisé.

Paramètres ajoutés Memcached 1.6.6

Pour Memcached 1.6.6, aucun paramètre supplémentaire n'est pris en charge.

Famille de groupes de paramètres : memcached1.6

Modifications des paramètres Memcached 1.5.10

Pour Memcached 1.5.10, les paramètres supplémentaires suivants sont pris en charge.

Famille de groupes de paramètres : memcached1.5

Nom	Détails	Description
`no_modern`	Valeur par défaut : 1 Type : valeur booléenne Modifiable : oui Valeurs autorisées : 0,1 Les modifications prennent effet : au lancement	Alias pour désactiver `slab_reassign` , `lru_maintainer_thread` `lru_segmented` , et `maxconns_fast` les commandes. Lorsque vous utilisez Memcached 1.5 ou version ultérieure, définit `no_modern` également le hash_algorithm sur. `jenkins` De plus, lors de l'utilisation de Memcached 1.5.10, `inline_ascii_reponse` est contrôlé par le paramètre. `parallelly` Cela signifie que s'il `no_modern` est désactivé, il `inline_ascii_reponse` est désactivé . À partir de la version 1.5.16 du moteur Memcached, le `inline_ascii_response` paramètre ne s'applique plus, donc le `no_modern` fait d'être activé ou désactivé n'a aucun effet sur. `inline_ascii_reponse`

Nom	Détails	Description
		S'`no_modern` il est désactivé `slab_reassign` , alors `lru_maintainer_thr ead` `lru_segmented` ,, et `maxconns_fast` SERA activé. Comme `hash_algorithm` les paramètres `slab_automove` et ne sont pas des paramètres SWITCH, leur réglage est basé sur les configurations du groupe de paramètres. Si vous souhaitez désactiver ce paramètre `no_modern` et y revenir`modern`, vous devez configurer un groupe de paramètre s personnalisé pour désactiver ce paramètre, puis redémarrer pour que ces modifications prennent effet. **Note** La valeur de configuration par défaut de ce paramètre est passée de 0 à 1 à compter du 20 août 2021. La valeur par défaut mise à jour sera automatiq uement sélectionnée par ElastiCache les nouveaux utilisateurs pour chaque région après le 20 août 2021. ElastiCache Les utilisateurs existants dans

Nom	Détails	Description
		les régions avant le 20 août 2021 doivent modifier manuellement leurs groupes de paramètre s personnalisés afin de prendre en compte cette nouvelle modification.
`inline_ascii_resp`	Par défaut : 0 Type : valeur booléenne Modifiable : oui Valeurs autorisées : 0,1 Les modifications prennent effet : au lancement	Stocke les nombres issus de la réponse `VALUE` dans un élément en utilisant jusqu'à 24 octets. Léger ralentissement pour les ensembles `get` et `faster` ASCII.

Pour Memcached 1.5.10, les paramètres suivants sont supprimés.

Nom	Détails	Description
`expirezero_does_not_evict`	Par défaut : 0 Type : valeur booléenne Modifiable : oui Valeurs autorisées : 0,1	N'est plus pris en charge dans cette version.

Nom	Détails	Description
	Les modifications prennent effet : au lancement	
`modern`	Valeur par défaut : 1 Type : valeur booléenne Modifiable : Oui (nécessite de relancer s'il est défini sur `no_modern`) Valeurs autorisées : 0,1 Les modifications prennent effet : au lancement	N'est plus pris en charge dans cette version. À partir de cette version, `no-modern` est activé par défaut à chaque démarrage et redémarrage.

Paramètres ajoutés Memcached 1.4.34

Pour Memcached 1.4.34, aucun paramètre supplémentaire n'est pris en charge.

Famille de groupes de paramètres : memcached1.4

Paramètres ajoutés Memcached 1.4.33

Pour Memcached 1.4.33, les paramètres supplémentaires suivants sont pris en charge.

Famille de groupes de paramètres : memcached1.4

Nom	Détails	Description
`modern`	Par défaut : activé	Un alias pour plusieurs fonctions . Activer `modern` équivaut à

Nom	Détails	Description
	Type : valeur booléenne Modifiable : oui Les modifications prennent effet : au lancement	activer les commandes suivantes et à utiliser un algorithme de hachage murmur3 : `slab_reassign` , `slab_automove` , `lru_crawler` , `lru_maintainer` , `maxconns_fast` et `hash_algorithm=murmur3` .
`watch`	Par défaut : activé Type : valeur booléenne Modifiable : oui Les modifications prennent effet : immédiatement Des journaux sont supprimés si l'utilisateur atteint ses limites `watcher_logbuf_size` et `worker_logbuf_size` .	Extractions, expulsions ou mutations de journaux. Quand, par exemple, un utilisateur active `watch`, il peut voir des journaux lorsque `get`, `set`, `delete` ou `update` a lieu.
`idle_timeout`	Valeurs : 0, 1. Type : entier Modifiable : oui Les modifications prennent effet : au lancement	Nombre minimum de secondes pendant lesquelles un client sera autorisé à rester inactif avant d'être invité à se fermer. Plage de valeurs : 0 à 86400.

Nom	Détails	Description
`track_sizes`	Par défaut : désactivé Type : valeur booléenne Modifiable : oui Les modifications prennent effet : au lancement	Affiche les tailles consommées par chaque groupe de sections. Activer `track_sizes` vous permet d'exécuter `stats sizes` sans avoir à exécuter `stats sizes_enable` .
`watcher_logbuf_size`	Défaut : 256 (Ko) Type : entier Modifiable : oui Les modifications prennent effet : au lancement	La commande `watch` active la journalisation des flux pour Memcached. Cependant `watch` peut supprimer des journaux si le taux d'expulsions, de mutations ou d'extraction est suffisamment élevé pour que la mémoire tampon du journal soit saturée. Dans de telles situations, les utilisateurs peuvent augmenter la taille de la mémoire tampon pour réduire les risques de perte de journaux.

Nom	Détails	Description
`worker_logbuf_size`	Défaut : 64 (Ko) Type : entier Modifiable : oui Les modifications prennent effet : au lancement	La commande `watch` active la journalisation des flux pour Memcached. Cependant `watch` peut supprimer des journaux si le taux d'expulsions, de mutations ou d'extraction est suffisamment élevé pour que la mémoire tampon du journal soit pleine. Dans de telles situations, les utilisateurs peuvent augmenter la taille de la mémoire tampon pour réduire les risques de perte de journaux.
`slab_chunk_max`	Par défaut : 524288 (octets) Type : entier Modifiable : oui Les modifications prennent effet : au lancement	Spécifie la taille maximale d'une section. La définition de sections plus petites permet d'utiliser plus efficacement la mémoire. Les éléments dont la taille est supérieure à `slab_chunk_max` sont répartis sur plusieurs sections.
`lru_crawler metadump [all|1|2|3]`	Par défaut : désactivé Type : valeur booléenne Modifiable : oui Les modifications prennent effet : immédiatement	Si lru_crawler est activé, cette commande vide toutes les clés. `all|1|2|3` - toutes les sections, ou spécifiez un numéro de section particulier

Paramètres ajoutés Memcached 1.4.24

Pour Memcached 1.4.24, les paramètres supplémentaires suivants sont pris en charge.

Famille de groupes de paramètres : memcached1.4

Nom	Détails	Description
`disable_flush_all`	Valeurs : 0, 1. Type : valeur booléenne Modifiable : oui Les modifications prennent effet : au lancement	Ajoutez un paramètre (`-F`) pour désactiver flush_all. Utile si vous ne souhaitez jamais effectuer de vidage complet de la mémoire sur les instances de production. Valeurs : 0, 1 (un utilisateur peut lancer un `flush_all` lorsque la valeur est 0).
`hash_algorithm`	Valeur par défaut : jenkins Type : chaîne Modifiable : oui Les modifications prennent effet : au lancement	L'algorithme de hachage à utiliser. Valeurs autorisées : murmur3 et jenkins.
`lru_crawler`	Valeurs : 0, 1. Type : valeur booléenne Modifiable : oui Les modifications prennent effet : après le redémarrage	Nettoie les classes de sections des éléments qui ont expiré. Il s'agit d'un processus de faible impact qui s'exécute en arrière-plan. Cela nécessite actuellem ent de lancer une analyse à l'aide d'une commande manuelle.

Nom	Détails	Description
	Note Vous pouvez activer temporairement `lru_crawler` lors de l'exécution de la ligne de commande. Pour plus d'informations, consultez la colonne Description.	Pour activer temporairement, exécutez `lru_crawler enable` à la ligne de commande. `lru_crawler 1,3,5` analyse les classes de sections 1, 3 et 5 à la recherche des éléments arrivés à expiration à ajouter à la freelist. Valeurs : 0,1 Note Activer `lru_crawler` à la ligne de commande permet d'activer le crawler jusqu'à ce qu'il soit désactivé à la ligne de commande ou au prochain redémarrage. Pour activer en permanence, vous devez modifier la valeur du paramètre. Pour de plus amples informations, veuillez consulter Modification d'un groupe ElastiCache de paramètres.

Nom	Détails	Description
`lru_maintainer`	Valeurs : 0, 1. Type : valeur booléenne Modifiable : oui Les modifications prennent effet : au lancement	Un fil d'arrière-plan qui mélange les éléments entre les deux au LRUs fur et à mesure que les capacités sont atteintes. Valeurs : 0, 1.
`expirezero_does_not_evict`	Valeurs : 0, 1. Type : valeur booléenne Modifiable : oui Les modifications prennent effet : au lancement	Si utilisé avec le paramètre `lru_maintainer` , les éléments ayant une date d'expiration de 0 ne peuvent pas être supprimés. Warning Cela évite d'utiliser la mémoire disponible pour les autres éléments pouvant être supprimés. Peut être paramétré pour ne pas tenir compte de `lru_maintainer` .

Paramètres ajoutés Memcached 1.4.14

Pour Memcached 1.4.14, les paramètres supplémentaires suivants sont pris en charge.

Famille de groupes de paramètres : memcached1.4

Paramètres ajoutés dans Memcached 1.4.14

Nom	Description
`config_max`	Le nombre maximum d'entrées de ElastiCache configuration.
`config_size_max`	La taille maximale des entrées configuration, en octets.
`hashpower_init`	La taille initiale de la table de ElastiCache hachage, exprimée sous la forme d'une puissance de deux. La valeur par défaut est 16 (2 ^ 16) ou 65536 clés.
`maxconns_fast`	Change la façon dont les nouvelles demandes de connexions sont gérées lorsque la limite de connexion maximale est atteinte. Si ce paramètre est défini sur 0 (zéro), les nouvelles connexions sont ajoutées

Nom	Description
	au journal des éléments en attente et attendront jusqu'à ce que les autres connexions soient terminées. Si le paramètre est défini sur 1, ElastiCache envoie une erreur au client et ferme immédiatement la connexion.
`slab_automove`	Ajuste l'algorithme slab automove : si ce paramètre est défini sur 0 (zéro), l'algorithme automove est désactivé. Si ce paramètre est défini sur 1, ElastiCache adopte une approche lente et prudente pour déplacer automatiquement les sections. S'il est réglé sur 2, déplace les dalles de ElastiCache manière agressive chaque fois qu'il y a une expulsion. (Ce mode n'est pas recommandé sauf pour effectuer des tests).

Nom	Description
`slab_reassign`	Activer ou désactiver la réaffectation de sections. Si ce paramètre est défini sur 1, vous pouvez utiliser la commande « slab_reassign » pour réaffecter manuellement la mémoire.

Paramètres Memcached 1.4.5 pris en charge

Famille de groupes de paramètres : memcached1.4

Pour Memcached 1.4.5, les paramètres suivants sont pris en charge.

Paramètres ajoutés dans Memcached 1.4.5

Nom	Détails	Description
`backlog_queue_limit`	Par défaut: 1024 Type : entier Modifiable : non	La limite de file d'attente de éléments en attente.

Nom	Détails	Description
`binding_protocol`	Valeur par défaut : auto Type : chaîne Modifiable : oui Les modifications prennent effet : après le redémarrage	Le protocole de liaison. Les valeurs autorisées sont : `ascii` et `auto`. Pour une assistance sur la modification de la valeur de `binding_protocol` , consultez Modification d'un groupe ElastiCache de paramètres.
`cas_disabled`	Par défaut : 0 (faux) Type : booléen Modifiable : oui Les modifications prennent effet : après le redémarrage	Si 1 (vrai), la vérification et la définition des opérations (CAS) seront désactivées, et les éléments stockés consommeront 8 fois moins d'octets que lorsque CAS est activé.
`chunk_size`	Par défaut: 48 Type : entier Modifiable : oui Les modifications prennent effet : après le redémarrage	Le montant minimum, en octets, de l'espace à allouer à la clé, à la valeur et aux indicateurs du plus petit élément.
`chunk_size_growth_factor`	Par défaut: 1.25 Type : décimal Modifiable : oui Les modifications prennent effet : après le redémarrage	Le facteur de croissance qui contrôle la taille de chaque segment Memcached successif ; chaque segment sera `chunk_size_growth_factor` fois plus important que le segment précédent.

Nom	Détails	Description
`error_on_memory_exhausted`	Par défaut : 0 (faux) Type : booléen Modifiable : oui Les modifications prennent effet : après le redémarrage	Si 1 (vrai), il n'y a pas plus de mémoire disponible pour stocker les éléments, Memcached retournera une erreur plutôt que d'expulser des éléments.
`large_memory_pages`	Par défaut : 0 (faux) Type : booléen Modifiable : non	Si 1 (vrai), j' ElastiCache essaierai d'utiliser de grandes pages de mémoire.
`lock_down_paged_memory`	Par défaut : 0 (faux) Type : booléen Modifiable : non	Si 1 (vrai), toutes les pages de mémoire ElastiCache seront verrouillées.
`max_item_size`	Par défaut: 1048576 Type : entier Modifiable : oui Les modifications prennent effet : après le redémarrage	La taille, en octets, de l'élément de plus grand qui peut être stocké dans le cluster.
`max_simultaneous_connections`	Par défaut: 65000 Type : entier Modifiable : non	Le nombre maximum de connexions simultanées.

Nom	Détails	Description
`maximize_core_file_limit`	Par défaut : 0 (faux) Type : booléen Adaptabilité: Les modifications prennent effet : après le redémarrage	Si 1 (vrai), ElastiCache maximisera la limite du fichier principal.
`memcached_connections_overhead`	Par défaut : 100 Type : entier Modifiable : oui Les modifications prennent effet : après le redémarrage	La quantité de mémoire à réserver pour les connexions Memcached et autre traitement divers. Pour plus d'informations sur ce paramètre, consultez Surcharge de la connexion Memcached.
`requests_per_event`	Valeur par défaut : 20 Type : entier Modifiable : non	Le nombre maximum de requêtes par événement pour une connexion donnée. Cette limite est nécessaire pour éviter une pénurie de ressources.

Surcharge de la connexion Memcached

Sur chaque nœud de , la mémoire mise à disposition pour stocker les éléments de est la mémoire disponible totale sur ce nœud (qui est stockée dans le paramètre `max_cache_memory`) moins la mémoire utilisée pour les connexions et autres traitements (qui est stockée dans le paramètre `memcached_connections_overhead`). Par exemple, un nœud de type `cache.m1.small` a un `max_cache_memory` de 1 300 Mo. Avec la valeur par défaut `memcached_connections_overhead` de 100 Mo, le processus de Memcached aura 1 200 Mo disponibles pour stocker les éléments.

Les valeurs par défaut du paramètre `memcached_connections_overhead` conviennent à la plupart des cas d'utilisation ; cependant, le nombre requis d'allocation au surcharge de connexion peut dépendre de plusieurs facteurs, comme le taux de demande, la taille de la charge utile et le nombre de connexions.

Vous pouvez modifier la valeur du paramètre `memcached_connections_overhead` afin de mieux correspondre aux besoins de votre application. Par exemple, l'augmentation de la valeur du paramètre `memcached_connections_overhead` réduira le volume de mémoire disponible pour stocker les éléments de et offrir une plus grande mémoire tampon pour la surcharge de connexion. Diminuer la valeur du paramètre `memcached_connections_overhead` vous permet d'avoir plus de mémoire pour stocker les éléments de , mais peut augmenter l'utilisation de l'espace d'échange et dégrader les performances. Si vous notez qu'il y a utilisation de l'espace d'échange et dégradation des performances, essayez d'augmenter la valeur du paramètre `memcached_connections_overhead`.

> **Important**
>
> Pour le type de nœud `cache.t1.micro`, la valeur du paramètre `memcached_connections_overhead` est déterminé comme suit :
>
> - Si votre cluster utilise le groupe de paramètres par défaut, la valeur ElastiCache sera définie `memcached_connections_overhead` sur 13 Mo.
> - Si votre cluster utilise un groupe de paramètres que vous avez créé, vous pouvez attribuer la valeur de votre choix au paramètre `memcached_connections_overhead`.

Paramètres propres au type de nœud Memcached

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 illustre les valeurs par défaut des paramètres `max_cache_memory` et `num_threads` pour chaque type de nœud. Les valeurs de ces paramètres ne peuvent pas être modifiées.

Type de nœud	max_cache_memory (en méga-octets)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1
cache.t2.medium	3301	2

Type de nœud	max_cache_memory (en méga-octets)	num_threads
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g. petit	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8

Type de nœud	max_cache_memory (en méga-octets)	num_threads
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16

Type de nœud	max_cache_memory (en méga-octets)	num_threads
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64

Type de nœud	max_cache_memory (en méga-octets)	num_threads
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

> **Note**
>
> Toutes les instances T2 sont créées dans un Amazon Virtual Private Cloud (Amazon VPC).

Connexion automatique d'une EC2 instance et d'un ElastiCache cache

Vous pouvez utiliser la ElastiCache console pour simplifier la configuration d'une connexion entre une instance Amazon Elastic Compute Cloud (Amazon EC2) et un ElastiCache cache. Souvent, votre cache se trouve dans un sous-réseau privé et votre EC2 instance dans un sous-réseau public au sein d'un VPC. Vous pouvez utiliser un client SQL sur votre EC2 instance pour vous connecter à votre ElastiCache cache. L' EC2 instance peut également exécuter des serveurs Web ou des applications qui accèdent à votre ElastiCache cache privé.

Rubriques

- Connectivité automatique avec une EC2 instance
- Affichage des ressources de calcul connectées

Connectivité automatique avec une EC2 instance

Lorsque vous configurez une connexion entre une EC2 instance et un ElastiCache cache, configure ElastiCache automatiquement le groupe de sécurité VPC pour EC2 votre instance et pour ElastiCache votre cache.

Les conditions requises pour connecter une EC2 instance à un ElastiCache cache sont les suivantes :

- L' EC2 instance doit exister dans le même VPC que le ElastiCache cache.

  Si aucune EC2 instance n'existe dans le même VPC, la console fournit un lien pour en créer une.
- L'utilisateur qui configure la connectivité doit être autorisé à effectuer les EC2 opérations Amazon suivantes. Ces autorisations sont généralement ajoutées aux EC2 comptes lors de leur création.

Pour plus d'informations sur EC2 les autorisations, consultez la section Octroi des autorisations requises pour les EC2 ressources Amazon.

- `ec2:AuthorizeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Lorsque vous configurez une connexion à une EC2 instance, ElastiCache agissez conformément à la configuration actuelle des groupes de sécurité associés au ElastiCache cache et à l' EC2 instance, comme décrit dans le tableau suivant.

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
Il existe un ou plusieurs groupes de sécurité associés au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité ne possède qu'une seule règle entrante dont le groupe de sécurité VPC de EC2 l'instance est la source.	Un ou plusieurs groupes de sécurité sont associés à l' EC2 instance dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité n'a qu'une seule règle sortante avec le groupe de sécurité VPC ElastiCache du cache comme source.	ElastiCache ne prend aucune mesure. Une connexion a déjà été configurée automatiquement entre l' EC2 instance et le ElastiCache cache. Comme une connexion existe déjà entre l' EC2 instance et le ElastiCache cache, les groupes de sécurité ne sont pas modifiés.

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . • Il existe un ou plusieurs groupes de sécurité associés au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Cependant, ElastiCache vous ne pouvez utiliser aucun de ces groupes de sécurité pour la connexion avec l' EC2 instance. ElastiCache ne peut pas utiliser un groupe de sécurité qui n'a pas de règle entrante avec le groupe de sécurité VPC de EC2 l'instance comme source. ElastiCache ne peut pas non plus utiliser un groupe de sécurité qui a été modifié. Des exemples	L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé à l' EC2 instance dont le nom correspond au modèle`ec2-elasticache-${ec2InstanceId}:${cacheId}` . • Un ou plusieurs groupes de sécurité sont associés à l' EC2 instance dont le nom correspond au modèle`ec2-elasticache-${ec2InstanceId}:${cacheId}` . Cependant , ElastiCache vous ne pouvez utiliser aucun de ces groupes de sécurité pour la connexion avec le ElastiCache cache. ElastiCache ne peut pas utiliser un groupe de sécurité qui n'a pas de règle sortante avec le groupe de sécurité VPC ElastiCache du cache comme source. ElastiCache ne peut pas non plus utiliser un groupe de sécurité qui a été modifié.	ELC action: create new security groups

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
de modifications incluent l'ajout d'une règle ou la modification du port d'une règle existante.		
Il existe un ou plusieurs groupes de sécurité associés au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité ne possède qu'une seule règle entrante dont le groupe de sécurité VPC de EC2 l'instance est la source.	Un ou plusieurs groupes de sécurité sont associés à l' EC2 instance dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Cependant, ElastiCache vous ne pouvez utiliser aucun de ces groupes de sécurité pour la connexion avec le ElastiCache cache. ElastiCache ne peut pas utiliser un groupe de sécurité qui n'a pas de règle sortante avec le groupe de sécurité VPC ElastiCache du cache comme source. ElastiCache ne peut pas non plus utiliser un groupe de sécurité qui a été modifié.	ELC action: create new security groups

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
Il existe un ou plusieurs groupes de sécurité associés au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité ne possède qu'une seule règle entrante dont le groupe de sécurité VPC de EC2 l'instance est la source.	Un groupe EC2 de sécurité valide existe pour la connexion , mais il n'est pas associé à l' EC2 instance. Le nom de ce groupe de sécurité correspond au modèle `ec2-elasticache-${ec2InstanceId}:${cacheId}` . Il n'a pas été modifié. Il n'a qu'une seule règle sortante avec le groupe de sécurité VPC theElastiCache du cache comme source.	ELC action: associate EC2 security group

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé au ElastiCache cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . • Il existe un ou plusieurs groupes de sécurité associés au ElastiCac he cache dont le nom correspond au modèle`elasticache-ec2-${cacheId}:${ec2InstanceId}` . Cependant, ElastiCache vous ne pouvez utiliser aucun de ces groupes de sécurité pour la connexion avec l' EC2 instance. ElastiCache ne peut pas utiliser un groupe de sécurité qui n'a pas de règle entrante avec le groupe de sécurité VPC de EC2 l'instance comme source. ElastiCache ne peut pas non plus utiliser le	Un ou plusieurs groupes de sécurité sont associés à l' EC2 instance dont le nom correspond au modèle`ec2-elasticache-${ec2InstanceId}:${cacheId}` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité n'a qu'une seule règle sortante avec le groupe de sécurité VPC ElastiCache du cache comme source.	ELC action: create new security groups

Configuration actuelle ElastiCache du groupe de sécurité	Configuration actuelle EC2 du groupe de sécurité	ElastiCache action
groupe de sécurité qui a été modifié.		

ElastiCacheaction : créer de nouveaux groupes de sécurité

ElastiCache prend les mesures suivantes :

- Crée un nouveau groupe de sécurité qui correspond au modèle `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Ce groupe de sécurité possède une règle entrante dont le groupe de sécurité VPC de EC2 l'instance est la source. Ce groupe de sécurité est associé au ElastiCache cache et permet à l' EC2 instance d'y accéder.
- Crée un nouveau groupe de sécurité qui correspond au modèle `elasticache-ec2-${cacheId}:${ec2InstanceId}`. Ce groupe de sécurité dispose d'une règle sortante avec le groupe de sécurité VPC ElastiCache du cache comme cible. Ce groupe de sécurité est associé à l' EC2 instance et permet à EC2 celle-ci d'envoyer du trafic vers le ElastiCache cache.

ElastiCache action : associer un groupe EC2 de sécurité

ElastiCacheassocie le groupe de EC2 sécurité existant valide à l' EC2 instance. Ce groupe de sécurité permet à l' EC2instance d'envoyer du trafic vers le ElastiCache cache.

Affichage des ressources de calcul connectées

Vous pouvez utiliser le AWS Management Console pour afficher les ressources de calcul connectées à un ElastiCache cache. Les ressources affichées comprennent les connexions de ressources de calcul qui ont été configurées automatiquement. Par exemple, vous pouvez autoriser une ressource de calcul à accéder manuellement à un cache en ajoutant une règle au groupe de sécurité VPC associé au cache. Ces ressources n'apparaîtront pas dans la liste des ressources informatiques connectées.

Pour qu'une ressource de calcul soit répertoriée, les mêmes conditions doivent s'appliquer que lors de la connexion automatique d'une EC2 instance et d'un ElastiCache cache.

Pour afficher les ressources de calcul connectées à un ElastiCache cache

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la
2. Dans le volet de navigation, choisissez Caches, puis choisissez un cache Valkey ou Redis OSS.
3. Dans l'onglet Connectivité et sécurité, consultez les ressources de calcul dans la section Configurer la connexion de calcul.

Dimensionnement ElastiCache

Vous pouvez adapter votre ElastiCache cache à vos besoins. Les caches sans serveur et les clusters conçus par nos soins offrent plusieurs options de mise à l'échelle.

Mise à l'échelle ElastiCache sans serveur

ElastiCache Serverless s'adapte automatiquement au trafic de votre charge de travail à mesure qu'il augmente ou diminue. Pour chaque cache ElastiCache sans serveur, suit ElastiCache en permanence l'utilisation des ressources telles que le processeur, la mémoire et le réseau. Lorsque l'une de ces ressources est limitée, ElastiCache Serverless évolue en ajoutant une nouvelle partition et en redistribuant les données vers la nouvelle partition, sans aucune interruption de service pour votre application. Vous pouvez surveiller les ressources consommées par votre cache en CloudWatch surveillant la `BytesUsedForCache` métrique pour le stockage des données du cache et `ElastiCacheProcessingUnits` (ECPU) pour l'utilisation du calcul.

Définition de limites de mise à l'échelle pour gérer les coûts

Vous pouvez choisir de configurer une utilisation maximale à la fois pour le stockage des données du cache et ECPU/second pour votre cache afin de contrôler les coûts du cache. Cela garantira que l'utilisation de votre cache ne dépassera jamais l'utilisation maximale configurée.

Si vous définissez un maximum de dimensionnement, votre application risque de connaître une baisse des performances du cache lorsque celui-ci atteint le maximum. Lorsque vous définissez un maximum de stockage de données en cache et que votre stockage de données en cache atteint le

maximum, ElastiCache les données de votre cache qui ont un Time-To-Live (TTL) défini commencent à être expulsées, en utilisant la logique LRU. Si aucune donnée ne peut être expulsée, les demandes d'écriture de données supplémentaires recevront un message d'erreur OOM (mémoire insuffisante). Lorsque vous définissez un ECPU/second maximum et que l'utilisation du calcul de votre charge de travail dépasse cette valeur, ElastiCache les demandes commencent à être limitées.

Si vous définissez une limite maximale sur `BytesUsedForCache` ou`ElastiCacheProcessingUnits`, nous vous recommandons vivement de configurer une CloudWatch alarme à une valeur inférieure à la limite maximale afin que vous soyez averti lorsque votre cache fonctionne à proximité de ces limites. Nous vous recommandons de définir une alarme à 75 % de la limite maximale que vous avez définie. Consultez la documentation pour savoir comment configurer les CloudWatch alarmes.

Pré-dimensionnement avec Serverless ElastiCache

ElastiCache Prédimensionnement sans serveur

Avec le prédimensionnement, également appelé préchauffage, vous pouvez définir des limites minimales prises en charge pour votre ElastiCache cache. Vous pouvez définir ces minimums pour les unités de ElastiCache traitement (ECPUs) par seconde ou pour le stockage des données. Cela peut être utile pour préparer les événements de mise à l'échelle prévus. Par exemple, si une société de jeux prévoit une multiplication par 5 du nombre de connexions dès la première minute de lancement de son nouveau jeu, elle peut préparer son cache pour cette augmentation significative de l'utilisation.

Vous pouvez effectuer un pré-dimensionnement à l'aide de la ElastiCache console, de la CLI ou de l'API. ElastiCache Serverless met à jour les ECPUs/second informations disponibles dans le cache dans les 60 minutes et envoie une notification d'événement lorsque la mise à jour de la limite minimale est terminée.

Comment fonctionne le pré-dimensionnement

Lorsque la limite minimale pour ECPUs/second le stockage des données est mise à jour via la console, la CLI ou l'API, cette nouvelle limite est disponible dans un délai d'une heure. ElastiCache Serverless prend en charge 30 Ko ECPUs/second sur un cache vide, et jusqu'à 90 000 en ECPUs/sec cas d'utilisation de la fonction Read from Replica. ElastiCache Serverless for Valkey 8.0 peut doubler le nombre de demandes prises en charge par seconde (RPS) toutes les 2 à 3 minutes, pour atteindre 5 millions de RPS par cache contre zéro en moins de 13 minutes, avec une latence de lecture constante inférieure à la milliseconde p50. Si vous prévoyez qu'un événement de

dimensionnement à venir pourrait dépasser ce taux, nous vous recommandons de fixer le minimum ECPUs/second au pic ECPUs/sec attendu au moins 60 minutes avant l'événement de pointe. Dans le cas contraire, l'application risque de connaître une latence élevée et un ralentissement des demandes.

Une fois la mise à jour de la limite minimale terminée, ElastiCache Serverless commencera à vous mesurer pour le nouveau minimum ECPUs par seconde ou le nouveau minimum de stockage. Cela se produit même si votre application n'exécute pas de requêtes dans le cache ou si votre utilisation du stockage de données est inférieure au minimum. Lorsque vous abaissez la limite minimale par rapport à son paramètre actuel, la mise à jour est immédiate, de sorte que ElastiCache Serverless commence immédiatement à mesurer la nouvelle limite minimale.

> **Note**
>
> - Lorsque vous définissez une limite d'utilisation minimale, celle-ci vous est facturée même si votre utilisation réelle est inférieure à la limite d'utilisation minimale. L'utilisation de l'ECPU ou du stockage de données qui dépasse la limite d'utilisation minimale est facturée au tarif normal. Par exemple, si vous définissez une limite d'utilisation minimale de 100 000 dollars, ECPUs/second vous serez facturé au moins 1,224$ par heure (en utilisant les prix de l'ECPU indiqués dans us-east-1), même si votre utilisation est inférieure à ce minimum défini.
> - ElastiCache Serverless prend en charge l'échelle minimale requise au niveau agrégé du cache. ElastiCache Serverless prend également en charge un maximum de 30 000 ECPUs/second par slot (90 000 ECPUs/second lors de l'utilisation de Read from Replica à l'aide de connexions READONLY). Il est recommandé que votre application veille à ce que la distribution des clés entre les emplacements Valkey ou Redis OSS et le trafic entre les clés soient aussi uniformes que possible.

Définition des limites de mise à l'échelle à l'aide de la console et AWS CLI

Définition des limites de dimensionnement à l'aide de la AWS console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez le moteur qui s'exécute sur le cache que vous souhaitez modifier.

3. La liste des caches exécutant le moteur choisi s'affiche.
4. Choisissez le cache à modifier en cliquant sur le bouton d'option situé à gauche du nom du cache.
5. Choisissez Actions, puis Modifier.
6. Sous Limites d'utilisation, définissez les limites de mémoire ou de calcul appropriées.
7. Cliquez sur Prévisualiser les modifications, puis sur Enregistrer les modifications.

Définition des limites d'échelle à l'aide du AWS CLI

Pour modifier les limites de dimensionnement à l'aide de la CLI, utilisez l' modify-serverless-cacheAPI.

Linux :

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
 ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},
 ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Suppression des limites de mise à l'échelle à l'aide de l'interface de ligne de commande

Pour supprimer les limites de mise à l'échelle à l'aide de la CLI, définissez les paramètres de limite minimale et maximale sur 0.

Linux :

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
 ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
```

```
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
 ECPUPerSecond={Minimum=0,Maximum=0}'
```

Mise à l'échelle de clusters conçus par vous-même

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 votre cache, vous devez mettre en service le matériel nécessaire pour répondre aux pics de demande, ce qui peut s'avérer coûteux. En utilisant Amazon, ElastiCache vous pouvez évoluer pour répondre à la demande actuelle, en ne payant que pour ce que vous utilisez. ElastiCache vous permet de dimensionner votre cache en fonction de la demande.

> **Note**
>
> Si un cluster Valkey ou Redis OSS est répliqué dans une ou plusieurs régions, ces régions sont redimensionnées dans l'ordre. Lors de la mise à l'échelle, les régions secondaires sont redimensionnées en premier, puis la région principale. Lors de la réduction, la région principale est la première, puis les régions secondaires suivent.
> Lors de la mise à jour de la version du moteur, la commande est la région secondaire, puis la région principale.

Rubriques

- Mise à l'échelle pour les clusters Memcached
- Mise à l'échelle manuelle pour Memcached
- Mise à l'échelle des clusters pour Valkey ou Redis OSS (mode cluster désactivé)
- Dimensionnement des nœuds de réplication pour Valkey ou Redis OSS (mode cluster désactivé)
- Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

Mise à l'échelle pour les clusters Memcached

ElastiCache for Memcached propose un service de mise en cache en mémoire entièrement géré qui déploie, exploite et fait évoluer verticalement Memcached dans le cloud. AWS

Dimensionnement vertical à la demande

Grâce à la mise à l'échelle verticale, ElastiCache Memcached fournit un système de mise en cache de mémoire distribuée à hautes performances largement utilisé pour accélérer les applications dynamiques en allégeant la charge des bases de données. Il stocke les données et les objets dans la RAM, réduisant ainsi le besoin de lire à partir de sources de données externes.

Vous pouvez utiliser des clusters conçus ElastiCache par Memcached pour une mise à l'échelle verticale. Vous pouvez appliquer une mise à l'échelle verticale aux clusters existants ainsi qu'aux nouveaux clusters. Cela peut apporter de la flexibilité dans l'allocation des ressources, permettant aux utilisateurs de s'adapter efficacement à l'évolution des charges de travail sans modifier l'architecture du cluster. Cette capacité d'évolutivité améliore les performances en augmentant la capacité du cache pendant les périodes de forte demande et en la réduisant pour optimiser les coûts pendant les périodes de faible demande. Cela simplifie les opérations, élimine le besoin de créer de nouveaux clusters pour répondre à l'évolution des besoins en ressources et permet de réagir rapidement aux fluctuations du trafic. Dans l'ensemble, la mise à l'échelle verticale des caches conçus par Memcached peut contribuer à améliorer la rentabilité, à améliorer l'utilisation des ressources et même à permettre aux utilisateurs de modifier leur type d'instance Memcached. Tout cela permet aux utilisateurs d'aligner plus facilement leur infrastructure de mise en cache sur les besoins réels des applications.

> **Note**
>
> - Les modifications de cache et de type de nœud conçues par vos soins ne sont disponibles que pour les versions 1.5 ou ultérieures du moteur Memcached.
> - La découverte automatique doit être activée pour pouvoir utiliser la mise à l'échelle verticale.

Configuration de la mise à l'échelle verticale à la demande pour les ElastiCache clusters Memcached

Vous pouvez configurer le dimensionnement vertical à la demande pour Memcached avec`scale-config`, qui contient deux paramètres :

1. ScaleIntervalMinutes: durée (en minutes) entre le dimensionnement des lots pendant le processus de mise à niveau de Memcached
2. ScalePercentage: pourcentage de nœuds à dimensionner simultanément pendant le processus de mise à niveau de Memcached

Conversion d'un type de nœud Memcached existant en un cache pouvant être redimensionné verticalement via la CLI

Pour convertir un cache personnalisé Memcached existant en un cache pouvant être redimensionné verticalement, vous pouvez l'utiliser via `elasticache modify-cache-cluster` la CLI.

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id <your-cluster-id> \
    --cache-node-type <new-node-type> \
    --scale-config <scale-config> \
    --apply-immediately
```

Configuration de la mise à l'échelle verticale avec la CLI

Pour configurer la mise à l'échelle verticale de votre cache personnalisé Memcached via la CLI, utilisez `elasticache modify-cache-cluster` with `scale-config` et ses paramètres et. `ScalePercentage` `ScaleIntervalMinutes`

- scale-interval-minutes: Ceci définit le temps (en minutes) entre les lots de mise à l'échelle. Ce réglage peut aller de 2 à 30 minutes. Si aucune valeur n'est spécifiée, la valeur par défaut de 5 minutes est appliquée.
- scale-percentage : indique le pourcentage de nœuds à dimensionner simultanément dans chaque lot. Ce paramètre peut être compris entre 10 et 100. Le paramètre est arrondi lors de la division. Par exemple, si le résultat est de 49,5, un paramètre de 50 est appliqué. Si aucune valeur n'est spécifiée, la valeur par défaut de 20 est appliquée.

Ces options de configuration vous permettront d'affiner le processus de dimensionnement en fonction de vos besoins spécifiques, en trouvant un équilibre entre la minimisation des perturbations du cluster et l'optimisation de la vitesse de mise à l'échelle. Le paramètre scale-config ne sera applicable qu'aux types de moteurs Memcached et sera ignoré pour les autres moteurs de cache, garantissant ainsi la rétrocompatibilité avec l'utilisation existante des API pour d'autres clusters.

Appel d'API

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id <your-cluster-id> \
    --cache-node-type <new-node-type> \
    --scale-config '{
            "ScalePercentage": 30,
```

```
            "ScaleIntervalMinutes": 2
          }'
    --apply-immediately
```

Résultat :

Renvoie l'ID du cluster de cache et la modification en attente.

```
{
    "CacheCluster": {
        "CacheNodeType": "old_insance_type",
         ...
         ...
         "PendingModifiedValues": {
            "CacheNodeType": "new_instance_type"
         },
    }
}
```

Répertoriez les paramètres de mise à l'échelle verticale de votre cache Memcached

Vous pouvez récupérer les options de mise à l'échelle de vos caches Memcached et voir quelles sont leurs options actuelles pour la mise à l'échelle verticale.

Appel d'API

```
aws elasticache list-allowed-node-type-modifications --cache-cluster-id <your-cluster-
id>
```

Résultat :

```
{
  "ScaleUpModifications": [
      "cache.x.xxxx",
      "cache.x.xxxx"
      ],
   "ScaleDownModifications": [
      "cache.x.xxxx",
      "cache.x.xxxx",
      "cache.x.xxxx"
      ]
```

```
}
```

Mise à l'échelle verticale pour Memcached avec AWS Management Console

Pour convertir un cache personnalisé Memcached existant en un cache pouvant être redimensionné verticalement à travers le https://console.aws.amazon.com/elasticache/, procédez comme suit.

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Sélectionnez le cache Memcached à convertir.
3. Sélectionnez l'onglet Modifier.
4. Accédez à la section Paramètres du cache et sélectionnez le type de nœud souhaité.
5. Sélectionnez Prévisualiser les modifications, puis passez en revue les modifications.
6. Sélectionnez Modify.

Mise à l'échelle horizontale automatisée pour Memcached

ElastiCache s'intègre désormais au service AWS Application Auto Scaling (AAS) pour inclure le dimensionnement horizontal automatique pour les clusters Memcached. Vous pouvez définir des politiques de dimensionnement via le service AWS Application Auto Scaling et ajuster automatiquement le nombre de nœuds dans les clusters Memcached selon les besoins, en fonction de métriques ou de plannings prédéfinis.

> **Note**
>
> La mise à l'échelle horizontale automatisée n'est actuellement pas disponible dans les régions de Pékin et de Ningxia.

Voici les méthodes disponibles pour redimensionner automatiquement horizontalement les caches que vous avez conçus vous-même.

- Dimensionnement planifié : le dimensionnement basé sur un calendrier vous permet de définir votre propre calendrier de dimensionnement pour des changements de charge prévisibles. Par exemple, chaque semaine le trafic vers l'application Web commence à augmenter le mercredi, reste élevé le jeudi et amorce une baisse le vendredi. Vous pouvez configurer Auto Scaling pour augmenter la capacité le mercredi et la diminuer le vendredi.

- Suivi des cibles : avec les politiques de dimensionnement du suivi des cibles, vous choisissez une métrique de mise à l'échelle et définissez une valeur cible. Application Auto Scaling crée et gère les CloudWatch alarmes qui déclenchent la politique de dimensionnement et calcule l'ajustement de dimensionnement en fonction de la métrique et de la valeur cible. La politique de dimensionnement ajoute ou supprime de la capacité si nécessaire pour maintenir la métrique à la valeur cible spécifiée ou proche de celle-ci.

Comment configurer la mise à l'échelle horizontale pour un cache auto-conçu ElastiCache pour Memcached via la CLI

Pour le dimensionnement horizontal avec ElastiCache Memcached, vous pouvez avoir une politique de suivi des cibles, une politique planifiée ou les deux.

1. Enregistrer une ressource en tant que cible évolutive

   Appelez l'`RegisterScalableTarget`API dans AWS Application Auto Scaling pour enregistrer la cible de la dimension évolutive`elasticache:cache-cluster:Nodes`.

   API : ApplicationAutoScaling. RegisterScalableTarget

   Entrée :

   ```
   {
    "ScalableDimension": "elasticache:cache-cluster:Nodes",
    "ResourceId": "cache-cluster/test-cluster-1",
    "ServiceNamespace": "elasticache",
    "MinCapacity": 20,
    "MaxCapacity": 50
   }
   ```

2. Création d'une politique de dimensionnement du suivi de Target

   Ensuite, vous pouvez créer une politique de dimensionnement de suivi des cibles pour la ressource en appelant l'API put scaling policy.

3. Métrique prédéfinie

   Vous trouverez ci-dessous une politique qui s'adapte à la dimension du nœud de cache, en utilisant la métrique prédéfinie `ElastiCacheCPUUtilization`, définie à 50 pour le cluster de cache test-cluster-1. Lorsque vous supprimez des nœuds à des fins de scale-in, les n derniers nœuds seront supprimés.

API : ApplicationAutoScaling. PutScalingPolicy

Entrée :

```
{
 "PolicyName": "cpu50-target-tracking-scaling-policy",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
  "TargetValue": 50,
  "PredefinedMetricSpecification": {
   "PredefinedMetricType": "ElastiCacheCPUUtilization"
   },
  "ScaleOutCooldown": 600,
  "ScaleInCooldown": 600
   },
 "ServiceNamespace": "elasticache",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ResourceId": "cache-cluster/test-cluster-1"
}
```

Sortie :

```
{
 "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/elasticache/cache-cluster/test-
cluster-1:policyName/cpu50-target-tracking-scaling-policy",
 "Alarms": [
  {
  "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-
elasticache/cache-cluster/test-cluster-1-AlarmHigh-d4f0770c-b46e-434a-
a60f-3b36d653feca",
  "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-AlarmHigh-
d4f0770c-b46e-434a-a60f-3b36d653feca"
  },
  {
  "AlarmARN": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:TargetTracking-
elasticache/cache-cluster/test-cluster-1-AlarmLow-1b437334-d19b-4a63-
a812-6c67aaf2910d",
  "AlarmName": "TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmLow-1b437334-d19b-4a63-a812-6c67aaf2910d"
  }
 ]
```

```
}
```

4. Métrique personnalisée

   Vous pouvez également définir une politique de dimensionnement pour la dimension en utilisant un pourcentage personnalisé basé sur la métrique Cloudwatch.

   Entrée :

```
{
 "PolicyName": "cpu50-target-tracking-scaling-policy",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
  "CustomizedMetricSpecification": {
   "Dimensions": [
    {
    "Name": "MyMetricDimension",
    "Value": "DimensionValue"
    }
    ],
   "MetricName": "MyCustomMetric",
   "Namespace": "MyNamespace",
   "Statistic": "Average",
   "Unit": "Percent"
   },
  "TargetValue": 40,
  "ScaleOutCooldown": 600,
  "ScaleInCooldown": 600
  },
 "ServiceNamespace": "elasticache",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ResourceId": "cache-cluster/test-cluster-1"
}
```

5. Actions planifiées

   Lorsque vous devez effectuer une mise à l'échelle en fonction d'un événement particulier, puis intervenir après l'événement, vous pouvez créer deux actions planifiées en appelant l'`PutScheduledAction`API.

   Politique 1 : Extensification

La `at` commande `--schedule` planifie l'action pour qu'elle soit exécutée une fois à une date et à une heure spécifiées dans le futur. Le champ de planification prend également en charge le taux (minute, heure, jour, etc.) et le cron (pour l'expression cron).

À la date et à l'heure spécifiées, Application Auto Scaling met à jour les valeurs `MinCapacity` et `MaxCapacity`. Application Auto Scaling s'adapte à 70 MinCapacity pour porter le nombre de nœuds de cache à 70.

API : ApplicationAutoScaling. PutScheduledAction

Entrée :

```
{
 "ResourceId": "elasticache:ache-cluster:test-cluster-1",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
  "ScalableTargetAction": {
   "MaxCapacity": 100,
   "MinCapacity": 70
   },
 "Schedule": "at(2020-05-20T17:05:00)",
 "ScheduledActionName": "ScalingOutScheduledAction",
 "ServiceNamespace": "elasticache",
}
```

Politique 2 : mise à l'échelle

À la date et à l'heure spécifiées, Application Auto Scaling met à jour les valeurs de `MinCapacity` la table et les redimensionne `MaxCapacity` pour ramener le nombre de nœuds de cache à 60. `MaxCapacity`

API : ApplicationAutoScaling. PutScheduledAction

Entrée :

```
{
 "ResourceId": "elasticache:cache-cluster:test-cluster-1",
 "ScalableDimension": "elasticache:cache-cluster:Nodes",
 "ScalableTargetAction": {
  "MaxCapacity": 60,
  "MinCapacity": 40
  },
```

```
 "Schedule": "at(2020-05-21T17:05:00)",
 "ScheduledActionName": "ScalingInScheduledAction",
 "ServiceNamespace": "elasticache",
}
```

6. Afficher les activités de mise à l'échelle

   Vous pouvez consulter les activités de dimensionnement à l'aide de l'`DescribeScalingActivities`API.

   API : ApplicationAutoScaling. DescribeScalingActivities

   Sortie :

```
{
 "ScalingActivities": [
  {
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 30.",
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "4d759079-a31f-4d0c-8468-504c56e2eecf",
  "StartTime": 1462574194.658,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574276.686,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy
 cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Failed to set desired count to 30",
  "StatusCode": "Failed"
  },
  {
  "ScalableDimension": "elasticache:elasticache:DesiredCount",
  "Description": "Setting desired count to 25.",
  "ResourceId": "elasticache/cache-cluster/test-cluster-1",
  "ActivityId": "90aff0eb-dd6a-443c-889b-b809e78061c1",
  "StartTime": 1462574254.223,
  "elasticacheNamespace": "elasticache",
  "EndTime": 1462574333.492,
  "Cause": "monitor alarm TargetTracking-elasticache/cache-cluster/test-cluster-1-
AlarmHigh-d4f0770c-b46e-434a-a60f-3b36d653feca in state ALARM triggered policy
 cpu50-target-tracking-scaling-policy",
  "StatusMessage": "Successfully set desired count to 25. Change successfully
 fulfilled by elasticache.",
```

```
  "StatusCode": "Successful"
  }
 ]
}
```

7. Modifier/supprimer la politique de dimensionnement

   Vous pouvez modifier ou supprimer des politiques en appelant à nouveau `PutScalingPolicy` l'API ou en appelant `DeleteScalingPolicy` `DeleteScheduled` Action.

8. Désenregistrer les cibles évolutives

   Vous pouvez désenregistrer la cible évolutive via l'`DeregisterScalableTarget`API. L'annulation de l'enregistrement d'une cible évolutive supprime les politiques de dimensionnement et les actions planifiées qui lui sont associées.

   API : ApplicationAutoScaling. DeregisterScalableTarget

   Entrée :

```
{
 "ResourceId": "elasticache/cache-cluster/test-cluster-1",
 "ServiceNamespace": "elasticache",
 "ScalableDimension": "elasticache:cache-cluster:Nodes"
}
```

9. Nettoyage des politiques de dimensionnement

10. Politiques de dimensionnement multiples

    Vous pouvez créer plusieurs politiques de dimensionnement. Vous trouverez ci-dessous des informations clés sur le comportement issues du suivi des cibles Auto Scaling.

    - Vous pouvez avoir plusieurs stratégies de dimensionnement Suivi de la cible pour une cible évolutive, dans la mesure où chacune d'elles utilise une métrique différente.
    - L'objectif d'Application Auto Scaling est de toujours donner la priorité à la disponibilité, afin que son comportement diffère selon que les politiques de suivi des cibles et d'échelonnement sont prêtes pour une augmentation ou une diminution de la capacité. Il augmentera la taille de la cible scalable si une des stratégies Suivi de la cible est prête pour une augmentation de taille, mais la diminuera uniquement si toutes les stratégies Suivi de la cible (avec la portion de diminution en charge activée) sont prêtes pour une diminution de taille.

- Si plusieurs stratégies indiquent simultanément à la cible évolutive de procéder à une montée en charge ou à une diminution en charge, Application Auto Scaling effectue la mise à l'échelle en fonction de la stratégie qui fournit la plus grande capacité à la fois pour la diminution et l'augmentation. Cela vous offre une plus grande flexibilité pour couvrir plusieurs scénarios et pouvoir toujours disposer d'une capacité suffisante pour traiter vos charges de travail d'application.

> **Note**
>
> AWS Application Auto Scaling ne met pas en file d'attente les politiques de dimensionnement. Application Auto Scaling attendra la fin de la première mise à l'échelle, puis se refroidira, puis répétera l'algorithme ci-dessus.

Redimensionnez automatiquement horizontalement un cache Memcached via AWS Management Console

Pour convertir un cache personnalisé Memcached existant en un cache pouvant être redimensionné horizontalement https://console.aws.amazon.com/elasticache/, procédez comme suit.

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Sélectionnez le cache Memcached à convertir.
3. Accédez à l'onglet Autoscaling.
4. Sélectionnez la politique de dimensionnement à appliquer en sélectionnant Ajouter une mise à l'échelle dynamique ou Ajouter une mise à l'échelle planifiée.
5. Renseignez les détails de la politique sélectionnée selon vos besoins.
6. Cliquez sur Create.

Mise à l'échelle manuelle pour Memcached

La mise à l'échelle horizontale manuelle d'un cluster Memcached vers ou vers l'extérieur est aussi simple que d'ajouter ou de supprimer des nœuds du cluster. Les clusters Memcached sont composés de 1 à 60 nœuds.

Etant donné que vous pouvez partitionner vos données sur tous les nœuds d'un cluster de Memcached, il est rarement obligatoire de changer pour un type de nœud avec une plus grande mémoire. Cependant, étant donné que le moteur Memcached ne conserve pas les données, si vous effectuez une mise à l'échelle vers un autre type de nœud, votre nouveau cluster repart vide à moins que votre application ne le remplisse.

Pour redimensionner manuellement verticalement votre cluster Memcached, vous devez créer un nouveau cluster. Les clusters de Memcached commencent toujours vides, sauf si votre application les remplisse.

Dimensionnement manuel des clusters Memcached

Action	Rubrique
Augmentation d'échelle	Ajout de nœuds à un cluster
Diminution d'échelle	Suppression de nœuds d'un cluster
Modification des types de nœuds	Dimensionnement vertical manuel de Memcached

Rubriques

- Dimensionnement horizontal manuel de Memcached
- Dimensionnement vertical manuel de Memcached

Dimensionnement horizontal manuel de Memcached

Le moteur Memcached prend en charge le partitionnement de vos données sur plusieurs nœuds. En conséquent, il est facile de dimensionner horizontalement les clusters de Memcached. Pour dimensionner horizontalement votre cluster de Memcached, ils suffit simplement d'ajouter ou de supprimer des nœuds.

Les rubriques suivantes expliquent en détail comment dimensionner votre cluster de Memcached ou en ajoutant ou supprimant des nœuds.

- Ajout de nœuds à un cluster

- Suppression de nœuds de votre cluster

Chaque fois que vous modifiez le nombre de nœuds dans votre cluster Memcached, vous devez redéfinir le mappage, au minimum, d'une partie de votre espace de clé vers le nœud adéquat. Pour en savoir plus sur l'équilibrage de charge de votre cluster Memcached, consultez Configuration de votre ElastiCache client pour un équilibrage de charge efficace (Memcached).

Si vous utilisez la découverte automatique sur votre cluster de Memcached, vous n'avez pas besoin de modifier les points de terminaison de votre application à mesure que vous ajoutez ou supprimez des nœuds. Pour plus d'informations sur la découverte automatique, consultez Identifiez automatiquement les nœuds de votre cluster (Memcached). Si vous n'utilisez pas la découverte automatique, chaque fois que vous modifiez le nombre de nœuds de votre cluster Memcached, vous devez mettre à jour les points de terminaison de votre application.

Dimensionnement vertical manuel de Memcached

Lorsque vous redimensionnez manuellement votre cluster Memcached vers le haut ou vers le bas, vous devez créer un nouveau cluster. Les clusters de Memcached commencent toujours vides, sauf si votre application les remplisse.

> **Important**
>
> Si vous optez pour un plus petit type de nœud, veillez à ce que le type de nœud plus petit soit suffisant pour vos données et les besoins de traitement. Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

Rubriques

- Mise à l'échelle verticale de Memcached (console)
- Mise à l'échelle verticale de Memcached (AWS CLI)
- Dimensionnement vertical de Memcached (API) ElastiCache

Mise à l'échelle verticale de Memcached (console)

La procédure suivante vous explique comment dimensionner verticalement votre cluster à l'aide de la ElastiCache console.

Pour mettre à l'échelle verticalement un cluster Memcached (console)

1. Créez un nouveau cluster de avec le nouveau type de nœud. Pour de plus amples informations, veuillez consulter Création d'un cluster Memcached (console).
2. Créez un nouveau cluster de cache avec le nouveau type de nœud. Pour de plus amples informations, veuillez consulter Trouver les points de terminaison d'un cluster (console) (Memcached).
3. Supprimez l'ancien cluster. Pour plus d'informations, consultez Suppression d'un nouveau nœud dans Memcached.

Mise à l'échelle verticale de Memcached (AWS CLI)

La procédure suivante vous guide à travers la mise à l'échelle verticale de votre cluster de cache Memcached à l'aide de l' AWS CLI.

Pour mettre à l'échelle verticalement un cluster de cache Memcached (AWS CLI)

1. Créez un nouveau cluster de cache avec le nouveau type de nœud. Pour de plus amples informations, veuillez consulter Création d'un cluster (AWS CLI).
2. Créez un nouveau cluster de cache avec le nouveau type de nœud. Pour de plus amples informations, veuillez consulter Recherche des points de terminaison (AWS CLI).
3. Supprimez l'ancien cluster de cache. Pour de plus amples informations, veuillez consulter Utilisation du AWS CLI pour supprimer un ElastiCache cluster.

Dimensionnement vertical de Memcached (API) ElastiCache

La procédure suivante vous explique comment dimensionner verticalement votre cluster de cache Memcached à l'aide de l' ElastiCache API.

Pour dimensionner verticalement un cluster de cache Memcached (API) ElastiCache

1. Créez un nouveau cluster de cache avec le nouveau type de nœud. Pour de plus amples informations, consultez Création d'un cluster pour Memcached (API) ElastiCache .
2. Procédez à la restauration à partir de l'instantané en utilisant le nouveau type de nœud plus petit comme type de nœud du cluster de cache et, si nécessaire, le nouveau groupe de paramètres. Pour de plus amples informations, veuillez consulter Recherche de points de terminaison (ElastiCache API).

3. Supprimez l'ancien cluster de cache. Pour de plus amples informations, veuillez consulter Utilisation de l' ElastiCache API.

Mise à l'échelle des clusters pour Valkey ou Redis OSS (mode cluster désactivé)

Les clusters Valkey ou Redis OSS (mode cluster désactivé) peuvent être des clusters à nœud unique avec 0 partition ou des clusters à nœuds multiples avec 1 partition. Les clusters à un seul nœud utilisent ce nœud pour se connecter aussi bien en écriture qu'en lecture. Les clusters à nœuds multiples ont toujours 1 nœud comme nœud read/write principal avec 0 à 5 nœuds de réplication en lecture seule.

Rubriques

- Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé)

Mise à l'échelle des clusters Valkey ou Redis OSS

Action	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
Diminution d'échelle	Supprimer des nœuds d'un ElastiCache cluster	Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)
Augmentation d'échelle	Ajout de nœuds à un cluster	Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)
Modification des types de nœuds	Vers un type de nœud plus grand • Mise à l'échelle de clusters Valkey ou Redis OSS à nœud unique • Augmenter les clusters Valkey ou Redis OSS avec des répliques Vers un type de nœud plus petit	Dimensionnement vertical en ligne en modifiant le type de nœud

Action	Valkey ou Redis OSS (mode cluster désactivé)	Valkey ou Redis OSS (mode cluster activé)
	• Réduction de la taille des clusters Valkey ou Redis OSS à nœud unique • Réduction de la taille des clusters Valkey ou Redis OSS avec des répliques	
Modification du nombre de groupes de nœuds	Non pris en charge pour les clusters Valkey ou Redis OSS (mode cluster désactivé)	Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

Table des matières

- Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé)
  - Mise à l'échelle de clusters Valkey ou Redis OSS à nœud unique
    - Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé) (console)
    - Mise à l'échelle de clusters de cache Valkey ou Redis OSS à nœud unique ()AWS CLI
    - Mise à l'échelle des clusters de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache
  - Réduction de la taille des clusters Valkey ou Redis OSS à nœud unique
    - Réduction de la taille d'un cluster Valkey ou Redis OSS à nœud unique (console)
    - Réduction des clusters de cache Valkey ou Redis OSS à nœud unique ()AWS CLI
    - Réduction des clusters de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé)

Les nœuds Valkey ou Redis OSS (mode cluster désactivé) doivent être suffisamment grands pour contenir toutes les données du cache plus la surcharge de Valkey ou Redis OSS. Pour modifier la capacité de données de votre cluster Valkey ou Redis OSS (mode cluster désactivé), vous devez effectuer une mise à l'échelle verticale, en passant à un type de nœud plus grand pour augmenter la capacité de données, ou en réduisant à un type de nœud plus petit pour réduire la capacité de données.

Le processus de mise à ElastiCache l'échelle est conçu pour faire de son mieux pour conserver vos données existantes et nécessite une réplication réussie de Valkey ou Redis OSS. Pour les clusters Valkey ou Redis OSS (mode cluster désactivé), nous recommandons de mettre suffisamment de mémoire à la disposition de Valkey ou Redis OSS.

Vous ne pouvez pas partitionner vos données sur plusieurs clusters Valkey ou Redis OSS (mode cluster désactivé). Toutefois, si vous devez uniquement augmenter ou diminuer la capacité de lecture de votre cluster, vous pouvez créer un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds de réplication et ajouter ou supprimer des répliques de lecture. Pour créer un cluster Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds de réplication en utilisant votre cluster de cache Valkey ou Redis OSS à nœud unique comme cluster principal, voir. Création d'un cluster Valkey (mode cluster désactivé) (console)

Une fois que vous avez créé le cluster avec réplicas, vous pouvez augmenter la capacité de lecture en ajoutant des réplicas en lecture. Plus tard, si nécessaire, vous pourrez réduire la capacité de lecture en éliminant les réplicas en lecture. Pour plus d'informations, consultez Augmentation de la capacité de lecture ou Diminution de la capacité de lecture.

Outre la possibilité d'augmenter la capacité de lecture, les clusters Valkey ou Redis OSS (mode cluster désactivé) dotés de répliques offrent d'autres avantages commerciaux. Pour de plus amples informations, veuillez consulter Haute disponibilité avec les groupes de réplication.

> **Important**
>
> Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, avant de commencer le dimensionnement, assurez-vous de disposer d'un groupe de paramètres personnalisé qui réserve la quantité de mémoire appropriée pour votre nouveau type de nœud. Vous pouvez aussi modifier un groupe de paramètres personnalisé de façon à ce qu'il utilise `reserved-memory-percent` et vous servir de ce groupe de paramètres pour votre nouveau cluster.
> Si vous utilisez `reserved-memory-percent`, cette opération n'est pas nécessaire.
> Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Rubriques

- Mise à l'échelle de clusters Valkey ou Redis OSS à nœud unique
- Réduction de la taille des clusters Valkey ou Redis OSS à nœud unique

Mise à l'échelle de clusters Valkey ou Redis OSS à nœud unique

Lorsque vous augmentez la taille d'un cluster Valkey ou Redis OSS à nœud unique, ElastiCache effectuez le processus suivant, que vous utilisiez la ElastiCache console, l'API ou l' AWS CLI API. ElastiCache

1. Un nouveau cluster de cache avec le nouveau type de nœud est exécuté dans la même zone de disponibilité que le cluster de cache existant.
2. Les données du cache dans le cluster de cache existant sont copiées dans le nouveau cluster de cache. La durée de ce processus dépend de votre type de nœud et de la quantité de données dans le cluster de cache.
3. Les opérations de lecture et d'écriture sont désormais réalisées avec le nouveau cluster de cache. Étant donné que les points de terminaison du nouveau cluster de cache sont identiques à ceux de l'ancien cluster, vous n'avez pas besoin de mettre à jour les points de terminaison dans votre application. Vous remarquerez une brève interruption des lectures et des écritures à partir du nœud primaire pendant que l'entrée DNS est mise à jour.
4. ElastiCache supprime l'ancien cluster de cache. Vous remarquerez une brève interruption (quelques secondes) des lectures et des écritures à partir de l'ancien nœud car les connexions à l'ancien nœud seront déconnectées.

> **Note**
>
> Pour les clusters exécutant le type de nœud r6gd, vous pouvez uniquement mettre à l'échelle en fonction des tailles de nœuds au sein de la famille de nœuds r6gd.

Comme indiqué dans le tableau suivant, votre opération de mise à l'échelle de Valkey ou Redis OSS est bloquée si une mise à niveau du moteur est prévue pour la prochaine fenêtre de maintenance. Pour plus d'informations, consultez Gestion de la maintenance des ElastiCache clusters.

Opérations Valkey ou Redis OSS bloquées

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

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

Si vous avez une opération en suspens qui vous bloque, vous pouvez effectuer l'une des actions suivantes.

- Planifiez votre opération de mise à l'échelle de Valkey ou Redis OSS pour la prochaine fenêtre de maintenance en décochant la case Appliquer immédiatement (utilisation de la CLI :`--no-apply-immediately`, utilisation de l'API :). `ApplyImmediately=false`
- Attendez votre prochaine fenêtre de maintenance (ou après) pour effectuer votre opération de mise à l'échelle de Valkey ou Redis OSS.
- Ajoutez la mise à niveau du moteur Valkey ou Redis OSS à cette modification du cluster de cache en cochant la case Appliquer immédiatement (utilisation de la CLI :`--apply-immediately`, utilisation de l'API :`ApplyImmediately=true`). Cela permet de débloquer votre opération de mise à l'échelle en provoquant une mise à jour du moteur à effectuer immédiatement.

Vous pouvez étendre un cluster Valkey ou Redis OSS à nœud unique (mode cluster désactivé) à l'aide de la ElastiCache console, de l'API ou de l'API AWS CLI. ElastiCache

> **Important**
>
> Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, avant de commencer le dimensionnement, assurez-vous de disposer d'un groupe de paramètres personnalisé qui réserve la quantité de mémoire appropriée pour votre nouveau type de nœud. Vous pouvez aussi modifier un groupe de paramètres personnalisé de façon à ce qu'il utilise `reserved-memory-percent` et vous servir de ce groupe de paramètres pour votre nouveau cluster.
> Si vous utilisez `reserved-memory-percent`, cette opération n'est pas nécessaire.
> Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Mise à l'échelle de clusters à nœud unique pour Valkey ou Redis OSS (mode cluster désactivé) (console)

La procédure suivante décrit comment augmenter la taille d'un cluster Valkey ou Redis OSS à nœud unique à l'aide de la ElastiCache console de gestion. Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour étendre un cluster Valkey ou Redis OSS à nœud unique (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey ou Redis OSS clusters.
3. Dans la liste des clusters, choisissez le cluster que vous souhaitez augmenter (il doit exécuter le moteur Valkey ou Redis OSS, et non le moteur Valkey ou Redis OSS en cluster).
4. Sélectionnez Modifier.
5. Dans l'assistant Modifier le cluster :
   a. Choisissez le type de nœud auquel vous souhaitez passer dans la liste Type de nœud.
   b. Si vous utilisez `reserved-memory` pour gérer la mémoire, dans la liste Groupe de paramètres, choisissez le groupe de paramètres personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud.
6. Si vous souhaitez effectuer un processus de mise à l'échelle immédiatement, cochez la case Appliquer immédiatement. Si la case Appliquer immédiatement est décochée, le processus de mise à l'échelle est effectué lors du créneau de maintenance suivant du cluster.
7. Sélectionnez Modifier.

   Si vous avez choisi Apply immediately lors de l'étape précédente, le statut du cluster passe à modification. Lorsque son statut passe à available, la modification est terminée et vous pouvez commencer à utiliser le nouveau cluster.

Mise à l'échelle de clusters de cache Valkey ou Redis OSS à nœud unique ()AWS CLI

La procédure suivante explique comment augmenter la taille d'un cluster de cache Valkey ou Redis OSS à nœud unique à l'aide du. AWS CLI Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour étendre un cluster de cache Valkey ou Redis OSS à nœud unique ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez augmenter en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

   - `--cache-cluster-id`

   Pour Linux, macOS ou Unix :

```
aws elasticache list-allowed-node-type-modifications \
     --cache-cluster-id my-cache-cluster-id
```

   Pour Windows :

```
aws elasticache list-allowed-node-type-modifications ^
     --cache-cluster-id my-cache-cluster-id
```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
     "ScaleUpModifications": [
         "cache.m3.2xlarge",
         "cache.m3.large",
         "cache.m3.xlarge",
         "cache.m4.10xlarge",
         "cache.m4.2xlarge",
         "cache.m4.4xlarge",
         "cache.m4.large",
         "cache.m4.xlarge",
         "cache.r3.2xlarge",
         "cache.r3.4xlarge",
         "cache.r3.8xlarge",
         "cache.r3.large",
         "cache.r3.xlarge"
     ]
        "ScaleDownModifications": [
         "cache.t2.micro",
         "cache.t2.small ",
         "cache.t2.medium ",
            "cache.t1.small ",
     ],
```

```
}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Modifiez votre cluster de cache existant en spécifiant le cluster de cache à augmenter et le nouveau type de nœud plus grand, à l'aide de la AWS CLI `modify-cache-cluster` commande et des paramètres suivants.

   - `--cache-cluster-id` : le nom du cluster de cache que vous êtes en train d'augmenter.
   - `--cache-node-type` : le nouveau type de nœud pour lequel vous voulez mettre à l'échelle le cluster de cache. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-modifications` lors de l'étape 1.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` : ce paramètre entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant du cluster, utilisez le paramètre `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
     --cache-cluster-id my-redis-cache-cluster \
     --cache-node-type cache.m3.xlarge \
     --cache-parameter-group-name redis32-m2-xl \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-cache-cluster ^
     --cache-cluster-id my-redis-cache-cluster ^
     --cache-node-type cache.m3.xlarge ^
     --cache-parameter-group-name redis32-m2-xl ^
     --apply-immediately
```

Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
     "CacheCluster": {
         "Engine": "redis",
         "CacheParameterGroup": {
             "CacheNodeIdsToReboot": [],
             "CacheParameterGroupName": "default.redis6.x",
             "ParameterApplyStatus": "in-sync"
         },
         "SnapshotRetentionLimit": 1,
         "CacheClusterId": "my-redis-cache-cluster",
         "CacheSecurityGroups": [],
         "NumCacheNodes": 1,
         "SnapshotWindow": "00:00-01:00",
         "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
         "AutoMinorVersionUpgrade": true,
         "CacheClusterStatus": "modifying",
         "PreferredAvailabilityZone": "us-west-2a",
         "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
         "CacheSubnetGroupName": "default",
         "EngineVersion": "6.0",
         "PendingModifiedValues": {
             "CacheNodeType": "cache.m3.2xlarge"
         },
         "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
         "CacheNodeType": "cache.m3.medium",
          "DataTiering": "disabled"
     }
 }
```

Pour plus d'informations, consultez modify-cache-cluster dans la référence AWS CLI .

3. Si vous avez utilisé le`--apply-immediately`, vérifiez l'état du nouveau cluster de cache à l'aide de la AWS CLI `describe-cache-clusters` commande avec le paramètre suivant. Lorsque le statut passe à available, vous pouvez commencer à utiliser le nouveau cluster de cache de taille supérieure.

   - `--cache-cache cluster-id`— Le nom de votre cluster de cache Valkey ou Redis OSS à nœud unique. Utilisez ce paramètre pour décrire un cluster de cache particulier plutôt que tous les clusters de cache.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Pour plus d'informations, consultez describe-cache-clusters dans la référence AWS CLI .

Mise à l'échelle des clusters de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

La procédure suivante décrit comment augmenter la taille d'un cluster de cache Valkey ou Redis OSS à nœud unique à l'aide de l'API. ElastiCache Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour étendre un cluster de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

1. Déterminez les types de nœuds que vous pouvez augmenter en exécutant l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `CacheClusterId`— Le nom du cluster de cache Valkey ou Redis OSS à nœud unique que vous souhaitez augmenter.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &CacheClusterId=MyRedisCacheCluster
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Modifiez votre cluster de cache existant en spécifiant le cluster de cache à augmenter et le nouveau type de nœud plus grand, à l'aide de l'action `ModifyCacheCluster` ElastiCache API et des paramètres suivants.

   - `CacheClusterId` : le nom du cluster de cache que vous êtes en train d'augmenter.

- `CacheNodeType` : le nouveau et le plus grand type de nœud pour lequel vous voulez mettre à l'échelle le cluster de cache. Cette valeur doit être l'un des types de nœuds renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.
- `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
- `ApplyImmediately` : lorsqu'il est défini sur `true`, il entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant du cluster, utilisez le paramètre `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &CacheClusterId=MyRedisCacheCluster
    &CacheNodeType=cache.m3.xlarge
    &CacheParameterGroupName redis32-m2-xl
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez ModifyCacheCluster le Amazon ElastiCache API Reference.

3. Si vous l'avez utilisé `ApplyImmediately=true`, vérifiez l'état du nouveau cluster de cache à l'aide de l'`DescribeCacheClusters`action ElastiCache API avec le paramètre suivant. Lorsque le statut passe à available, vous pouvez commencer à utiliser le nouveau cluster de cache de taille supérieure.

   - `CacheClusterId`— Le nom de votre cluster de cache Valkey ou Redis OSS à nœud unique. Utilisez ce paramètre pour décrire un cluster de cache particulier plutôt que tous les clusters de cache.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=MyRedisCacheCluster
```

```
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeCacheClusters le Amazon ElastiCache API Reference.

Réduction de la taille des clusters Valkey ou Redis OSS à nœud unique

Les sections suivantes vous expliquent comment réduire un cluster Valkey ou Redis OSS à nœud unique à un type de nœud plus petit. Pour garantir le succès à long terme de votre nouveau cluster Valkey ou Redis OSS, il est important de s'assurer que le nouveau type de nœud plus petit est suffisamment grand pour accueillir toutes les données et les surcharges liées à Valkey ou Redis OSS. Pour de plus amples informations, veuillez consulter S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS.

> **Note**
>
> Pour les clusters exécutant le type de nœud r6gd, vous pouvez uniquement mettre à l'échelle en fonction des tailles de nœuds au sein de la famille de nœuds r6gd.

Rubriques

- Réduction de la taille d'un cluster Valkey ou Redis OSS à nœud unique (console)
- Réduction des clusters de cache Valkey ou Redis OSS à nœud unique ()AWS CLI
- Réduction des clusters de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

Réduction de la taille d'un cluster Valkey ou Redis OSS à nœud unique (console)

La procédure suivante vous explique comment réduire votre cluster Valkey ou Redis OSS à nœud unique à un type de nœud plus petit à l'aide de la console. ElastiCache

> **Important**
>
> Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, avant de commencer le dimensionnement, assurez-vous de disposer d'un groupe de paramètres personnalisé qui réserve la quantité de mémoire appropriée pour votre nouveau type de nœud. Vous pouvez aussi modifier un groupe de paramètres personnalisé de façon à ce qu'il utilise `reserved-memory-percent` et vous servir de ce groupe de paramètres pour votre nouveau cluster.
> Si vous utilisez `reserved-memory-percent`, cette opération n'est pas nécessaire.
> Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Pour réduire la taille de votre cluster Valkey ou Redis OSS à nœud unique (console)

1. Assurez-vous que le type de nœud plus petit soit suffisant pour supporter vos données et les besoins de traitement.
2. Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, assurez-vous de disposer d'un groupe de paramètres personnalisé pour réserver la quantité de mémoire appropriée pour votre nouveau type de nœud.

   Vous pouvez aussi modifier votre groupe de paramètres personnalisé de façon à utiliser `reserved-memory-percent`. Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.
3. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
4. Dans la liste des clusters, choisissez le cluster que dont vous souhaitez baisser la capacité. Ce cluster doit exécuter le moteur Valkey ou Redis OSS et non le moteur Valkey ou Redis OSS en cluster.
5. Sélectionnez Modifier.
6. Dans l'assistant Modifier le cluster :

   a. Choisissez le type de nœud dont vous souhaitez baisser la capacité dans la liste Type de nœud.

   b. Si vous utilisez `reserved-memory` pour gérer la mémoire, dans la liste Groupe de paramètres, choisissez le groupe de paramètres personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud.
7. Si vous souhaitez effectuer le processus de diminution de la capacité immédiatement, choisissez la case Apply immediately. Si la case Apply immediately est décochée, le processus de diminution de la capacité est effectué lors du créneau de maintenance suivant du cluster.
8. Sélectionnez Modifier.
9. Lorsque le statut du cluster passe de modifying à available, cela signifie que votre cluster est passé au nouveau type de nœud. Il n'est pas nécessaire de mettre à jour les points de terminaison dans votre application.

Réduction des clusters de cache Valkey ou Redis OSS à nœud unique ()AWS CLI

La procédure suivante décrit comment réduire la taille d'un cluster de cache Valkey ou Redis OSS à nœud unique à l'aide du. AWS CLI

Pour réduire la taille d'un cluster de cache Valkey ou Redis OSS à nœud unique ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez réduire en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

   - `--cache-cluster-id`

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache list-allowed-node-type-modifications \
        --cache-cluster-id my-cache-cluster-id
   ```

   Pour Windows :

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --cache-cluster-id my-cache-cluster-id
   ```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
            "cache.r3.xlarge"
        ]
           "ScaleDownModifications": [
   ```

```
        "cache.t2.micro",
        "cache.t2.small ",
        "cache.t2.medium ",
           "cache.t1.small ",
    ],

}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Modifiez votre cluster de cache existant en spécifiant le cluster de cache à réduire et le nouveau type de nœud plus petit, à l'aide de la AWS CLI `modify-cache-cluster` commande et des paramètres suivants.

   - `--cache-cluster-id` : nom du cluster de cache que vous êtes en train de réduire.
   - `--cache-node-type` : le nouveau type de nœud pour lequel vous voulez mettre à l'échelle le cluster de cache. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-modifications` lors de l'étape 1.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` : entraîne l'application immédiate du processus de réduction. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant du cluster, utilisez le paramètre `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
     --cache-cluster-id my-redis-cache-cluster \
     --cache-node-type cache.m3.xlarge \
     --cache-parameter-group-name redis32-m2-xl \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-cache-cluster ^
     --cache-cluster-id my-redis-cache-cluster ^
     --cache-node-type cache.m3.xlarge ^
     --cache-parameter-group-name redis32-m2-xl ^
     --apply-immediately
```

Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
     "CacheCluster": {
         "Engine": "redis",
         "CacheParameterGroup": {
             "CacheNodeIdsToReboot": [],
             "CacheParameterGroupName": "default.redis6,x",
             "ParameterApplyStatus": "in-sync"
         },
         "SnapshotRetentionLimit": 1,
         "CacheClusterId": "my-redis-cache-cluster",
         "CacheSecurityGroups": [],
         "NumCacheNodes": 1,
         "SnapshotWindow": "00:00-01:00",
         "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
         "AutoMinorVersionUpgrade": true,
         "CacheClusterStatus": "modifying",
         "PreferredAvailabilityZone": "us-west-2a",
         "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
         "CacheSubnetGroupName": "default",
         "EngineVersion": "6.0",
         "PendingModifiedValues": {
             "CacheNodeType": "cache.m3.2xlarge"
         },
         "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
         "CacheNodeType": "cache.m3.medium",
          "DataTiering": "disabled"
     }
 }
```

Pour plus d'informations, consultez modify-cache-cluster dans la référence AWS CLI .

3. Si vous avez utilisé le `--apply-immediately`, vérifiez l'état du nouveau cluster de cache à l'aide de la AWS CLI `describe-cache-clusters` commande avec le paramètre suivant.

Lorsque le statut passe à available, vous pouvez commencer à utiliser le nouveau cluster de cache de taille supérieure.

- `--cache-cache cluster-id`— Le nom de votre cluster de cache Valkey ou Redis OSS à nœud unique. Utilisez ce paramètre pour décrire un cluster de cache particulier plutôt que tous les clusters de cache.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Pour plus d'informations, consultez describe-cache-clusters dans la référence AWS CLI .

Réduction des clusters de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

La procédure suivante décrit comment redimensionner un cluster de cache Valkey ou Redis OSS à nœud unique à l'aide de l'API. ElastiCache

Pour réduire la taille d'un cluster de cache (API) Valkey ou Redis OSS à nœud unique ElastiCache

1. Déterminez les types de nœuds que vous pouvez réduire en exécutant l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `CacheClusterId`— Le nom du cluster de cache Valkey ou Redis OSS à nœud unique que vous souhaitez réduire.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ListAllowedNodeTypeModifications
    &CacheClusterId=MyRedisCacheCluster
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

   Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Modifiez votre cluster de cache existant en spécifiant le cluster de cache à augmenter et le nouveau type de nœud plus grand, à l'aide de l'action `ModifyCacheCluster` ElastiCache API et des paramètres suivants.

   - `CacheClusterId` – Nom du cluster de cache que vous êtes en train de réduire.
   - `CacheNodeType` : nouveau et plus grand type de nœud vers lequel vous souhaitez mettre à l'échelle le cluster de cache. Cette valeur doit être l'un des types de nœuds renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.
   - `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `ApplyImmediately` : définissez ce paramètre sur `true` pour effectuer immédiatement le processus de réduction. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant du cluster, utilisez le paramètre `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=true
    &CacheClusterId=MyRedisCacheCluster
    &CacheNodeType=cache.m3.xlarge
    &CacheParameterGroupName redis32-m2-xl
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

   Pour plus d'informations, consultez ModifyCacheCluster le Amazon ElastiCache API Reference.

3. Si vous l'avez utilisé `ApplyImmediately=true`, vérifiez l'état du nouveau cluster de cache à l'aide de l'`DescribeCacheClusters`action ElastiCache API avec le paramètre suivant. Lorsque le statut passe à available (disponible), vous pouvez commencer à utiliser le nouveau cluster de cache plus petit.

- `CacheClusterId`— Le nom de votre cluster de cache Valkey ou Redis OSS à nœud unique. Utilisez ce paramètre pour décrire un cluster de cache particulier plutôt que tous les clusters de cache.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterId=MyRedisCacheCluster
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeCacheClusters le Amazon ElastiCache API Reference.

Dimensionnement des nœuds de réplication pour Valkey ou Redis OSS (mode cluster désactivé)

Un cluster Valkey ou Redis OSS doté de nœuds de réplication (appelé groupe de réplication dans l'API/CLI) assure une haute disponibilité grâce à une réplication dans laquelle le mode multi-AZ avec basculement automatique est activé. Un cluster avec des nœuds répliques est un ensemble logique de six nœuds Valkey ou Redis OSS au maximum dont un nœud, le nœud principal, est capable de répondre à la fois aux demandes de lecture et d'écriture. Tous les autres nœuds du cluster sont des réplicas en lecture seule du cluster principal. Les données écrites dans le cluster principal sont répliquées de façon asynchrone sur tous les réplicas en lecture du cluster. Valkey ou Redis OSS (mode cluster désactivé) ne prenant pas en charge le partitionnement de vos données sur plusieurs clusters, chaque nœud d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) contient l'intégralité du jeu de données de cache. Les clusters Valkey ou Redis OSS (mode cluster activé) permettent de partitionner vos données sur un maximum de 500 partitions.

Pour modifier la capacité de données de votre cluster, vous devez dimensionner le type de nœud en passant à un type de nœud plus grand ou plus petit.

Pour modifier la capacité de lecture de votre cluster, ajoutez des réplicas en lecture, jusqu'à un maximum de 5, ou supprimez-en.

Le processus de mise à ElastiCache l'échelle est conçu pour faire de son mieux pour conserver vos données existantes et nécessite une réplication réussie de Valkey ou Redis OSS. Pour les clusters Valkey ou Redis OSS dotés de répliques, nous recommandons de mettre suffisamment de mémoire à la disposition de Valkey ou Redis OSS.

Rubriques

- Augmenter les clusters Valkey ou Redis OSS avec des répliques
- Réduction de la taille des clusters Valkey ou Redis OSS avec des répliques
- Augmentation de la capacité de lecture
- Diminution de la capacité de lecture

Rubriques connexes

- Haute disponibilité avec les groupes de réplication
- Réplication : mode cluster Valkey et Redis OSS désactivé ou activé
- Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS

- S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS

Rubriques

- Augmenter les clusters Valkey ou Redis OSS avec des répliques
- Réduction de la taille des clusters Valkey ou Redis OSS avec des répliques
- Augmentation de la capacité de lecture
- Diminution de la capacité de lecture

Augmenter les clusters Valkey ou Redis OSS avec des répliques

Amazon ElastiCache fournit un support de console, de CLI et d'API pour dimensionner votre groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) vers le haut.

Lorsque le processus de mise à l'échelle est lancé, ElastiCache effectue les opérations suivantes :

1. Il lance un groupe de réplication à l'aide du nouveau type de nœud.
2. Il copie toutes les données du nœud principal actuel vers le nouveau nœud principal.
3. Il synchronise les nouveaux réplicas en lecture avec le nouveau nœud principal.
4. Il met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Ainsi, vous n'aurez plus besoin de mettre à jour les points de terminaison de votre application. Pour Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster continue de rester en ligne et de traiter les demandes entrantes. Sur les versions 4.0.10 et antérieures de Redis OSS, vous remarquerez peut-être une brève interruption des lectures et des écritures sur les versions précédentes depuis le nœud principal pendant la mise à jour de l'entrée DNS.
5. Il supprime les anciens nœuds (CLI/API : groupe de réplication). Vous remarquerez une brève interruption (quelques secondes) des lectures et des écritures à partir des anciens nœuds car les connexions aux anciens nœuds seront déconnectées.

La durée de ce processus dépend de votre type de nœud et de la quantité de données dans votre cluster.

Comme indiqué dans le tableau suivant, votre opération de mise à l'échelle de Valkey ou Redis OSS est bloquée si une mise à niveau du moteur est planifiée pour la prochaine fenêtre de maintenance du cluster.

Opérations Valkey ou Redis OSS bloquées

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

Si vous avez une opération en suspens qui vous bloque, vous pouvez effectuer l'une des actions suivantes.

- Planifiez votre opération de mise à l'échelle de Valkey ou Redis OSS pour la prochaine fenêtre de maintenance en décochant la case Appliquer immédiatement (utilisation de la CLI :`--no-apply-immediately`, utilisation de l'API :). `ApplyImmediately=false`
- Attendez votre prochaine fenêtre de maintenance (ou après) pour effectuer votre opération de mise à l'échelle de Valkey ou Redis OSS.
- Ajoutez la mise à niveau du moteur Valkey ou Redis OSS à cette modification du cluster de cache en cochant la case Appliquer immédiatement (utilisation de la CLI :`--apply-immediately`, utilisation de l'API :`ApplyImmediately=true`). Cela permet de débloquer votre opération de mise à l'échelle en provoquant une mise à jour du moteur à effectuer immédiatement.

Les sections suivantes décrivent comment redimensionner votre cluster Valkey ou Redis OSS avec des répliques à l'aide de la ElastiCache console, de l'API et de l' AWS CLI API. ElastiCache

> **Important**
>
> Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, avant de commencer le dimensionnement, assurez-vous de disposer d'un groupe de paramètres personnalisé qui réserve la quantité de mémoire appropriée pour votre nouveau type de nœud. Vous pouvez aussi modifier un groupe de paramètres personnalisé de façon à ce qu'il utilise `reserved-memory-percent` et vous servir de ce groupe de paramètres pour votre nouveau cluster.
> Si vous utilisez `reserved-memory-percent`, cette opération n'est pas nécessaire.
> Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Mise à l'échelle d'un cluster Valkey ou Redis OSS avec des répliques (console)

La durée nécessaire pour redimensionner la taille d'un type de nœud et passer à un type plus grand, varie selon le type de nœud et la quantité de données dans votre cluster de actuel.

Le processus suivant fait évoluer votre cluster avec des répliques de son type de nœud actuel vers un nouveau type de nœud plus grand à l'aide de la ElastiCache console. Pendant ce processus, une brève interruption des lectures et des écritures peut avoir lieu pour d'autres versions à partir du nœud

primaire pendant que l'entrée DNS est mise à jour. Vous pouvez constater un temps d'arrêt inférieur à une seconde pour les nœuds s'exécutant sur les versions 5.0.6 et de l'ordre de quelques secondes pour les versions plus anciennes.

Pour augmenter le cluster Valkey ou Redis OSS avec des répliques (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters
3. Dans la liste des clusters, choisissez le cluster que vous souhaitez augmenter. Ce cluster doit exécuter le moteur Valkey ou Redis OSS et non le moteur Valkey ou Redis OSS en cluster.
4. Sélectionnez Modifier.
5. Dans l'assistant Modifier le cluster :
   a. Choisissez le type de nœud auquel vous souhaitez passer dans la liste Type de nœud. Notez que tous les types de nœuds ne sont pas disponibles pour la réduction de la capacité.
   b. Si vous utilisez `reserved-memory` pour gérer la mémoire, dans la liste Groupe de paramètres, choisissez le groupe de paramètres personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud.
6. Si vous souhaitez effectuer un processus de mise à l'échelle immédiatement, choisissez la case Apply immediately. Si la case Apply immediately est décochée, le processus de mise à l'échelle est effectué lors du créneau de maintenance suivant du cluster.
7. Sélectionnez Modifier.
8. Lorsque le statut du cluster passe de modifying à available, cela signifie que votre cluster est passé au nouveau type de nœud. Il n'est pas nécessaire de mettre à jour les points de terminaison dans votre application.

Mise à l'échelle d'un groupe de réplication Valkey ou Redis OSS ()AWS CLI

Le processus suivant met à l'échelle votre groupe de réplication à partir de son type de nœud actuel vers un nouveau type de nœud plus grand à l'aide de l' AWS CLI. Au cours de ce processus, ElastiCache met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Ainsi, vous n'aurez plus besoin de mettre à jour les points de terminaison de votre application. Pour Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster continue de rester en ligne et de traiter les demandes entrantes. Sur les versions 4.0.10 ou antérieures, vous pouvez

remarquer une brève interruption des lectures et des écritures sur les versions précédentes à partir du nœud primaire pendant que l'entrée DNS est mise à jour.

La durée nécessaire pour remettre à l'échelle la taille d'un type de nœud et passer à un type plus grand, varie selon le type de nœud et la quantité de données dans votre cluster de actuel.

Pour agrandir un groupe de réplication Valkey ou Redis OSS ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez augmenter en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

   - `--replication-group-id` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-repl-group
   ```

   Pour Windows :

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-repl-group
   ```

   Le résultat de cette opération doit ressembler à ce qui suit (format JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
            "cache.r3.large",
   ```

```
        "cache.r3.xlarge"
    ]
}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Adaptez votre groupe de réplication actuel au nouveau type de nœud à l'aide de la AWS CLI `modify-replication-group` commande avec les paramètres suivants.

   - `--replication-group-id` : le nom du groupe de réplication.
   - `--cache-node-type` : le nouveau type de nœud plus grand des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par la `list-allowed-node-type-modifications` commande à l'étape précédente.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` : ce paramètre entraîne l'application immédiate du processus d'augmentation. Pour reporter l'opération de mise à l'échelle au créneau de maintenance suivant, utilisez `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
     --replication-group-id my-repl-group \
     --cache-node-type cache.m3.xlarge \
     --cache-parameter-group-name redis32-m3-2xl \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-replication-group ^
     --replication-group-id my-repl-group ^
     --cache-node-type cache.m3.xlarge ^
     --cache-parameter-group-name redis32-m3-2xl \
     --apply-immediately
```

Le résultat de cette commande doit ressembler à ce qui suit (format JSON).

```
{
 "ReplicationGroup": {
  "Status": "available",
  "Description": "Some description",
  "NodeGroups": [{
   "Status": "available",
   "NodeGroupMembers": [{
     "CurrentRole": "primary",
     "PreferredAvailabilityZone": "us-west-2b",
     "CacheNodeId": "0001",
     "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
     },
     "CacheClusterId": "my-repl-group-001"
    },
    {
     "CurrentRole": "replica",
     "PreferredAvailabilityZone": "us-west-2c",
     "CacheNodeId": "0001",
     "ReadEndpoint": {
      "Port": 6379,
      "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
     },
     "CacheClusterId": "my-repl-group-002"
    }
   ],
   "NodeGroupId": "0001",
   "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
   }
  }],
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",
  "SnapshottingClusterId": "my-repl-group-002",
  "MemberClusters": [
   "my-repl-group-001",
```

```
    "my-repl-group-002"
   ],
   "PendingModifiedValues": {}
  }
 }
```

Pour plus d'informations, consultez modify-replication-group dans la référence AWS CLI .

3. Si vous avez utilisé le `--apply-immediately` paramètre, surveillez l'état du groupe de réplication à l'aide de la AWS CLI `describe-replication-group` commande avec le paramètre suivant. Pendant que le statut est encore modification en cours, vous pouvez constater un temps d'arrêt inférieur à 1 seconde pour les nœuds s'exécutant sur les versions 5.0.6 et une brève interruption des lectures et des écritures pour les versions plus anciennes à partir du nœud primaire pendant que l'entrée DNS est mise à jour.

   - `--replication-group-id` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

   Pour Linux, macOS ou Unix :

```
aws elasticache describe-replication-groups \
     --replication-group-id my-replication-group
```

   Pour Windows :

```
aws elasticache describe-replication-groups ^
     --replication-group-id my-replication-group
```

   Pour plus d'informations, consultez describe-replication-groups dans la référence AWS CLI .

Mise à l'échelle d'un groupe de réplication (ElastiCache API) Valkey ou Redis OSS

Le processus suivant fait passer votre groupe de réplication de son type de nœud actuel à un nouveau type de nœud plus important à l'aide de l' ElastiCache API. Pour Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster continue de rester en ligne et de traiter les demandes entrantes. Sur les versions Redis OSS 4.0.10 et antérieures,

vous remarquerez peut-être une brève interruption des lectures et des écritures sur les versions précédentes depuis le nœud principal pendant la mise à jour de l'entrée DNS.

La durée nécessaire pour remettre à l'échelle la taille d'un type de nœud et passer à un type plus grand, varie selon le type de nœud et la quantité de données dans votre cluster de actuel.

Pour étendre un groupe de réplication (ElastiCache API) Valkey ou Redis OSS

1. Déterminez les types de nœuds que vous pouvez augmenter à l'aide de l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `ReplicationGroupId` : le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication spécifique plutôt que tous les groupes de réplication.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &ReplicationGroupId=MyReplGroup
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Adaptez votre groupe de réplication actuel au nouveau type de nœud à l'aide de l'action `ModifyReplicationGroup` ElastiCache API et avec les paramètres suivants.

   - `ReplicationGroupId` : le nom du groupe de réplication.
   - `CacheNodeType` : le nouveau type de nœud plus grand des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.
   - `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.

- `ApplyImmediately` : lorsqu'il est défini sur `true`, il entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant, utilisez `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez ModifyReplicationGroup le Amazon ElastiCache API Reference.

3. Si vous l'avez utilisé `ApplyImmediately=true`, surveillez l'état du groupe de réplication à l'aide de l'`DescribeReplicationGroups`action ElastiCache API avec les paramètres suivants. Lorsque le statut passe de modifying à available, cela signifie que vous pouvez commencer à écrire sur votre nouveau groupe de réplication redimensionné.

   - `ReplicationGroupId` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
```

```
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeReplicationGroups le Amazon ElastiCache API Reference.

Réduction de la taille des clusters Valkey ou Redis OSS avec des répliques

Les sections suivantes vous expliquent comment redimensionner un cluster de cache Valkey ou Redis OSS (mode cluster désactivé) avec des nœuds de réplication jusqu'à un type de nœud plus petit. Pour des raisons de performances, il faut s'assurer que le nouveau type de nœud plus petit a une capacité suffisante pour supporter toutes les données et les besoins de traitement. Pour de plus amples informations, veuillez consulter S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS.

> **Note**
>
> Pour les clusters exécutant le type de nœud r6gd, vous pouvez uniquement mettre à l'échelle en fonction des tailles de nœuds au sein de la famille de nœuds r6gd.

> **Important**
>
> Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, avant de commencer le dimensionnement, assurez-vous de disposer d'un groupe de paramètres personnalisé qui réserve la quantité de mémoire appropriée pour votre nouveau type de nœud. Vous pouvez aussi modifier un groupe de paramètres personnalisé de façon à ce qu'il utilise `reserved-memory-percent` et vous servir de ce groupe de paramètres pour votre nouveau cluster.
> Si vous utilisez `reserved-memory-percent`, cette opération n'est pas nécessaire.
> Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

Réduction de la taille d'un cluster Valkey ou Redis OSS avec des répliques (console)

Le processus suivant fait évoluer votre cluster Valkey ou Redis OSS avec des nœuds répliques vers un type de nœud plus petit à l'aide de la ElastiCache console.

Pour réduire la taille d'un cluster Valkey ou Redis OSS avec des nœuds de réplication (console)

1. Assurez-vous que le type de nœud plus petit soit suffisant pour supporter vos données et les besoins de traitement.
2. Si votre groupe de paramètres est utilisé `reserved-memory` pour réserver de la mémoire pour la surcharge de Valkey ou Redis OSS, assurez-vous de disposer d'un groupe de paramètres

personnalisé pour réserver la quantité de mémoire appropriée pour votre nouveau type de nœud.

Vous pouvez aussi modifier votre groupe de paramètres personnalisé de façon à utiliser `reserved-memory-percent`. Pour de plus amples informations, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.

3. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
4. Dans la liste des clusters, choisissez le cluster que dont vous souhaitez baisser la capacité. Ce cluster doit exécuter le moteur Valkey ou Redis OSS et non le moteur Valkey ou Redis OSS en cluster.
5. Sélectionnez Modifier.
6. Dans l'assistant Modifier le cluster :
   a. Choisissez le type de nœud dont vous souhaitez baisser la capacité dans la liste Type de nœud.
   b. Si vous utilisez `reserved-memory` pour gérer la mémoire, dans la liste Groupe de paramètres, choisissez le groupe de paramètres personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud.
7. Si vous souhaitez effectuer le processus de diminution de la capacité immédiatement, choisissez la case Apply immediately. Si la case Apply immediately est décochée, le processus de diminution de la capacité est effectué lors du créneau de maintenance suivant du cluster.
8. Sélectionnez Modifier.
9. Lorsque le statut du cluster passe de modifying à available, cela signifie que votre cluster est passé au nouveau type de nœud. Il n'est pas nécessaire de mettre à jour les points de terminaison dans votre application.

Réduction de la taille d'un groupe de réplication Valkey ou Redis OSS ()AWS CLI

Le processus suivant met à l'échelle votre groupe de réplication à partir de son type de nœud actuel vers un nouveau type de nœud plus petit à l'aide de l' AWS CLI. Au cours de ce processus, ElastiCache met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Ainsi, vous n'aurez plus besoin de mettre à jour les points de terminaison de votre application. Pour Valkey 7.2 ou Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster reste en ligne et traite les demandes entrantes. Sur les versions 4.0.10 ou antérieures, vous pouvez remarquer une brève interruption des lectures et

des écritures sur les versions précédentes à partir du nœud primaire pendant que l'entrée DNS est mise à jour.

Toutefois, les lectures des clusters de cache des réplicas en lecture continuent sans interruption.

La durée nécessaire à la réduction vers un type de nœud plus petit varie selon le type de nœud et la quantité de données dans votre cluster de cache actuel.

Pour réduire un groupe de réplication Valkey ou Redis OSS ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez réduire en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

   - `--replication-group-id` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-repl-group
   ```

   Pour Windows :

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-repl-group
   ```

   Le résultat de cette opération doit ressembler à ce qui suit (format JSON).

   ```
   {
        "ScaleDownModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
            "cache.r3.8xlarge",
   ```

```
        "cache.r3.large",
        "cache.r3.xlarge"
    ]
}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Adaptez votre groupe de réplication actuel au nouveau type de nœud à l'aide de la AWS CLI `modify-replication-group` commande avec les paramètres suivants.

   - `--replication-group-id` : le nom du groupe de réplication.
   - `--cache-node-type` : le nouveau type de nœud plus petit des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par la `list-allowed-node-type-modifications` commande à l'étape précédente.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` : ce paramètre entraîne l'application immédiate du processus d'augmentation. Pour reporter l'opération de mise à l'échelle au créneau de maintenance suivant, utilisez `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
     --replication-group-id my-repl-group \
     --cache-node-type cache.t2.small  \
     --cache-parameter-group-name redis32-m3-2xl \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-replication-group ^
     --replication-group-id my-repl-group ^
     --cache-node-type cache.t2.small  ^
     --cache-parameter-group-name redis32-m3-2xl \
```

```
     --apply-immediately
```

Le résultat de cette commande doit ressembler à ce qui suit (format JSON).

```
{"ReplicationGroup": {
         "Status": "available",
         "Description": "Some description",
         "NodeGroups": [
             {
                 "Status": "available",
                 "NodeGroupMembers": [
                     {
                         "CurrentRole": "primary",
                         "PreferredAvailabilityZone": "us-west-2b",
                         "CacheNodeId": "0001",
                         "ReadEndpoint": {
                             "Port": 6379,
                             "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
                         },
                         "CacheClusterId": "my-repl-group-001"
                     },
                     {
                         "CurrentRole": "replica",
                         "PreferredAvailabilityZone": "us-west-2c",
                         "CacheNodeId": "0001",
                         "ReadEndpoint": {
                             "Port": 6379,
                             "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
                         },
                         "CacheClusterId": "my-repl-group-002"
                     }
                 ],
                 "NodeGroupId": "0001",
                 "PrimaryEndpoint": {
                     "Port": 6379,
                     "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
                 }
             }
         ],
```

```
        "ReplicationGroupId": "my-repl-group",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "disabled",
        "SnapshotWindow": "12:00-13:00",
        "SnapshottingClusterId": "my-repl-group-002",
        "MemberClusters": [
            "my-repl-group-001",
            "my-repl-group-002",
        ],
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations, consultez modify-replication-group dans la référence AWS CLI .

3. Si vous avez utilisé le `--apply-immediately` paramètre, surveillez l'état du groupe de réplication à l'aide de la AWS CLI `describe-replication-group` commande avec le paramètre suivant. Lorsque le statut passe de modification à disponible, vous pouvez commencer à écrire dans votre nouveau groupe de réplication réduit.

   - `--replication-group-id` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

   Pour Linux, macOS ou Unix :

```
aws elasticache describe-replication-group \
     --replication-group-id my-replication-group
```

   Pour Windows :

```
aws elasticache describe-replication-groups ^
     --replication-group-id my-replication-group
```

   Pour plus d'informations, consultez describe-replication-groups dans la référence AWS CLI .

Réduction de la taille d'un groupe de réplication (ElastiCache API) Valkey ou Redis OSS

Le processus suivant fait passer votre groupe de réplication de son type de nœud actuel à un nouveau type de nœud plus petit à l'aide de l' ElastiCache API. Au cours de ce processus, ElastiCache met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Ainsi, vous

n'aurez plus besoin de mettre à jour les points de terminaison de votre application. Pour Valkey 7.2 et versions ultérieures ou Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster continue de rester en ligne et de traiter les demandes entrantes. Sur Redis OSS version 4.0.10 et antérieures, vous remarquerez peut-être une brève interruption des lectures et des écritures sur les versions précédentes depuis le nœud principal pendant la mise à jour de l'entrée DNS. Toutefois, les lectures des clusters de cache des réplicas en lecture continuent sans interruption.

La durée nécessaire à la réduction vers un type de nœud plus petit varie selon le type de nœud et la quantité de données dans votre cluster de cache actuel.

Pour réduire un groupe de réplication (ElastiCache API) Valkey ou Redis OSS

1. Déterminez les types de nœuds que vous pouvez réduire à l'aide de l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `ReplicationGroupId` : le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication spécifique plutôt que tous les groupes de réplication.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &ReplicationGroupId=MyReplGroup
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Adaptez votre groupe de réplication actuel au nouveau type de nœud à l'aide de l'action `ModifyReplicationGroup` ElastiCache API et avec les paramètres suivants.

   - `ReplicationGroupId` : le nom du groupe de réplication.
   - `CacheNodeType` : le nouveau type de nœud plus petit des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.

- `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
- `ApplyImmediately` – Lorsqu'il est défini sur `true`, il entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de réduction à la prochaine fenêtre de maintenance, utilisez `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez ModifyReplicationGroup le Amazon ElastiCache API Reference.

3. Si vous l'avez utilisé `ApplyImmediately=true`, surveillez l'état du groupe de réplication à l'aide de l'`DescribeReplicationGroups`action ElastiCache API avec les paramètres suivants. Lorsque le statut passe de modification à disponible, vous pouvez commencer à écrire dans votre nouveau groupe de réplication réduit.

   - `ReplicationGroupId` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeReplicationGroups le Amazon ElastiCache API Reference.

Augmentation de la capacité de lecture

Pour augmenter la capacité de lecture, ajoutez des répliques de lecture (jusqu'à cinq au maximum) à votre groupe de réplication Valkey ou Redis OSS.

Vous pouvez augmenter la capacité de lecture de votre cluster Valkey ou Redis OSS à l'aide de la ElastiCache console, de l'API AWS CLI ou de l' ElastiCache API. Pour de plus amples informations, veuillez consulter Ajouter une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé).

Diminution de la capacité de lecture

Pour réduire la capacité de lecture, supprimez une ou plusieurs répliques de lecture de votre cluster Valkey ou Redis OSS contenant des répliques (appelé groupe de réplication dans l'API/CLI). Si le mode Multi-AZ avec basculement automatique est activé pour ce cluster, vous ne pouvez pas supprimer le dernier réplica en lecture sans désactiver au préalable le mode Multi-AZ. Pour de plus amples informations, veuillez consulter Modification d'un groupe de réplication.

Pour de plus amples informations, veuillez consulter Suppression d'une réplique de lecture pour Valkey ou Redis OSS (mode cluster désactivé).

Mise à l'échelle des clusters dans Valkey ou Redis OSS (mode cluster activé)

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 dans votre cluster Valkey ou Redis OSS (mode cluster activé). 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 BytesUsedForCache.
- Goulet d'étranglement UC ou réseau :

  Si latency/throughput des problèmes affectent votre cluster, vous devrez peut-être le redimensionner pour les résoudre.

  Vous pouvez surveiller vos niveaux de latence et de débit en surveillant les métriques suivantes : CPUUtilization, NetworkBytesIn, NetworkBytesOutCurrConnections, et 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 : FreeableMemorySwapUsageBytesUsedForCache, CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections, 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 Valkey ou Redis OSS (mode cluster activé) : mise à l'échelle horizontale et verticale.

- Le dimensionnement horizontal vous permet de modifier le nombre de groupes de nœuds (partitions) dans le groupe de réplication en ajoutant ou en supprimant des groupes de nœuds (partitions). Le processus de repartage en ligne permet une mise à l'échelle in/out pendant que le cluster continue de traiter les demandes entrantes.

  Configurer les emplacements de votre nouveau cluster différemment qu'ils ne l'étaient dans l'ancien cluster. Méthode hors ligne uniquement.
- Dimensionnement vertical : modifier le type de nœud pour redimensionner le cluster. La mise à l'échelle verticale en ligne permet une mise à l'échelle up/down pendant que le cluster continue de traiter les 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 Valkey ou Redis OSS.

Pour de plus amples informations, veuillez consulter Choix de la taille de votre nœud.

Table des matières

- Repartage hors ligne pour Valkey ou Redis OSS (mode cluster activé)
- Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)
  - Ajout de partitions avec le repartitionnement en ligne
  - Suppression de partitions avec le repartitionnement en ligne
    - Suppression de partitions (console)
    - Suppression de partitions (AWS CLI)
    - Supprimer des fragments (ElastiCacheAPI)

- Rééquilibrage en ligne des partitions
  - Rééquilibrage en ligne des partitions (console)
  - Rééquilibrage des partitions en ligne (AWS CLI)
  - Rééquilibrage des partitions en ligne (API) ElastiCache
- Dimensionnement vertical en ligne en modifiant le type de nœud
  - Augmentation en ligne
    - Mise à l'échelle des clusters de cache Valkey ou Redis OSS (console)
    - Mise à l'échelle des clusters de cache Valkey ou Redis OSS ()AWS CLI
    - Mise à l'échelle des clusters de cache (ElastiCache API) Valkey ou Redis OSS
  - Réduction en ligne
    - Réduction de la taille des clusters de cache Valkey ou Redis OSS (console)
    - Réduction de la taille des clusters de cache Valkey ou Redis OSS ()AWS CLI
    - Réduction de la taille des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Repartage hors ligne pour Valkey ou Redis OSS (mode cluster activé)

Le principal avantage de la reconfiguration des partitions hors ligne est que vous pouvez faire bien plus qu'ajouter ou supprimer simplement des partitions de votre groupe de réplication. Lorsque vous redimensionnez et rééquilibrez hors ligne, en plus de modifier le nombre de partitions dans votre groupe de réplication, vous pouvez effectuer les opérations suivantes :

> **Note**
>
> Le repartage hors ligne n'est pas pris en charge sur les clusters Valkey ou Redis OSS avec la hiérarchisation des données activée. Pour de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

- Modifier le type de nœud de votre groupe de réplication.
- Spécifier la zone de disponibilité de chaque nœud du groupe de réplication.
- Mettre à niveau vers une version plus récente du moteur.
- Spécifier le nombre de nœuds de réplicas dans chaque partition indépendamment.
- Spécifiez l'espace de clés de chaque partition.

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 de partitions Valkey ou Redis OSS (mode cluster activé) hors ligne

1. Créez une sauvegarde manuelle de votre cluster Valkey ou Redis OSS existant. Pour de plus amples informations, veuillez consulter Réalisation de sauvegardes manuelles.
2. Créez un nouveau cluster en restaurant la sauvegarde. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.
3. Mettez à jour les points de terminaison dans votre application sur les points de terminaison du nouveau cluster. Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)

En utilisant le repartage en ligne et le rééquilibrage des partitions avec ElastiCache Valkey 7.2 ou version ultérieure, ou Redis OSS version 3.2.10 ou ultérieure, vous pouvez faire évoluer votre cluster Valkey ou Redis OSS (mode cluster activé) 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 :

- Dimensionnement : augmentez la capacité de lecture et d'écriture en ajoutant des fragments (groupes de nœuds) à votre cluster Valkey ou Redis OSS (mode cluster activé) (groupe de réplication).

  Si vous ajoutez une ou plusieurs partitions à votre groupe de réplication, le nombre de nœuds de chaque nouvelle partition est identique au nombre de nœuds de la plus petite des partitions existantes.
- Évolutivité : réduisez la capacité de lecture et d'écriture, et donc les coûts, en supprimant les fragments de votre cluster Valkey ou Redis OSS (mode cluster activé).
- Rééquilibrage — Déplacez les espaces clés entre les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) afin qu'ils soient répartis de la manière la plus égale possible entre les partitions.

Vous ne pouvez pas exécuter les actions suivantes :

- Configurer les partitions indépendamment :

  Vous ne pouvez pas spécifier l'espace de clés des partitions indépendamment. Pour ce faire, vous devez utiliser le processus hors ligne.

Actuellement, les restrictions suivantes s'appliquent au repartage et au rééquilibrage ElastiCache en ligne :

- Ces processus nécessitent Valkey 7.2 ou version ultérieure ou Redis OSS 3.2.10 ou version ultérieure. Pour plus d'informations sur la mise à niveau de la version de votre moteur, consultez Gestion des versions pour ElastiCache.
- 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 l'agrandissement, le nombre de nœuds des nouvelles partitions est égal au nombre de nœuds de la plus petite partition existante.
- Lors de la réduction, les balises communes à toutes les partitions existantes sont copiées sur les nouvelles partitions.
- Lors de la mise à l'échelle d'un cluster Global Data Store, les fonctions ne ElastiCache seront pas automatiquement répliquées depuis l'un des nœuds existants vers le ou les nouveaux nœuds. Nous vous recommandons de charger vos fonctions dans la ou les nouvelles partitions après avoir fait monter en puissance votre cluster afin que toutes les partitions aient les mêmes fonctions.

> **Note**
>
> ElastiCache Pour Valkey 7.2 et versions ultérieures, et ElastiCache pour Redis OSS version 7 et ultérieure : lors de la mise à l'échelle de votre cluster, les fonctions chargées dans l'un des nœuds existants (sélectionnées au hasard) ElastiCache seront automatiquement répliquées sur le ou les nouveaux nœuds. Si votre application utilise des fonctions, nous

vous recommandons de charger toutes vos fonctions sur toutes les partitions avant de les redimensionner afin que votre cluster ne se retrouve pas avec des définitions de fonctions différentes sur différentes partitions.

Pour de plus amples informations, veuillez consulter Redimensionnement de cluster en ligne.

Vous pouvez redimensionner ou rééquilibrer horizontalement vos clusters Valkey ou Redis OSS (mode cluster activé) à l'aide de l'API AWS Management Console, du et de l' AWS CLI API. ElastiCache

Ajout de partitions avec le repartitionnement en ligne

Vous pouvez ajouter des partitions à votre cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de l'API AWS Management Console AWS CLI, ou. ElastiCache Lorsque vous ajoutez des partitions à un cluster Valkey ou Redis OSS (mode cluster activé), toutes les balises des partitions existantes sont copiées sur les nouvelles partitions.

Ajout de partitions (console)

Vous pouvez utiliser le AWS Management Console pour ajouter une ou plusieurs partitions à votre cluster Valkey ou Redis OSS (mode cluster activé). La procédure suivante décrit le processus.

Pour ajouter des partitions à votre cluster Valkey ou Redis OSS (mode cluster activé)

1. Ouvrez la ElastiCache console à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Recherchez et choisissez le nom, et non la case à gauche du nom du cluster, du cluster Valkey ou Redis OSS (mode cluster activé) auquel vous souhaitez ajouter des partitions.

   > **Tip**
   >
   > Valkey ou Redis OSS (mode cluster activé) Afficher Clusterisé Valkey ou Clusterisé Redis OSS dans la colonne Mode

4. Choisissez Ajouter une partition.

   a. Pour Nombre de partitions à ajouter, choisissez le nombre de partitions que vous voulez ajouter au cluster.

b. Pour Zones de disponibilité, choisissez Aucune préférence ou Spécifier les zones de disponibilité.
c. Si vous choisissez Spécifier les zones de disponibilité, pour chaque nœud de chaque partition, sélectionnez la zone de disponibilité du nœud dans la liste Zones de disponibilité.
d. Choisissez Ajouter.

Ajout de partitions (AWS CLI)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en ajoutant des partitions à l'aide du. AWS CLI

Utilisez les paramètres suivants avec `modify-replication-group-shard-configuration`.

Paramètres

- `--apply-immediately` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.
- `--replication-group-id` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.
- `--node-group-count` – Obligatoire. Spécifie le nombre de partitions (groupes de nœuds) qui doivent exister une fois l'opération terminée. Lors de l'ajout des partitions, la valeur de `--node-group-count` doit être supérieure au nombre actuel de partitions.

  Le cas échéant, vous pouvez spécifier la zone de disponibilité de chaque nœud du groupe de réplication avec `--resharding-configuration`.
- `--resharding-configuration` : facultatif. Liste des zones de disponibilité favorites de chaque nœud de chaque partition du groupe de réplication. N'utilisez ce paramètre que si la valeur de `--node-group-count` est supérieure au nombre actuel de partitions. Si ce paramètre est omis lors de l'ajout de partitions, Amazon ElastiCache sélectionne les zones de disponibilité pour les nouveaux nœuds.

L'exemple suivant reconfigure les espaces clés sur quatre partitions dans un cluster Valkey ou Redis OSS (mode cluster activé) nommé. `my-cluster` L'exemple spécifie aussi la zone de disponibilité de chaque nœud de chaque partition. L'opération commence immédiatement.

Example - Ajout de partitions

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 4 \
    --resharding-configuration \
        "PreferredAvailabilityZones=us-east-2a,us-east-2c" \
        "PreferredAvailabilityZones=us-east-2b,us-east-2a" \
        "PreferredAvailabilityZones=us-east-2c,us-east-2d" \
        "PreferredAvailabilityZones=us-east-2d,us-east-2c" \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group-shard-configuration ^
    --replication-group-id my-cluster ^
    --node-group-count 4 ^
    --resharding-configuration ^
        "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^
        "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^
        "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^
        "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^
    --apply-immediately
```

Pour plus d'informations, consultez modify-replication-group-shard-configuration dans la AWS CLI documentation.

Ajouter des partitions (ElastiCache API)

Vous pouvez utiliser l' ElastiCache API pour reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en ligne en utilisant l'opération. `ModifyReplicationGroupShardConfiguration`

Utilisez les paramètres suivants avec `ModifyReplicationGroupShardConfiguration`.

Paramètres

- `ApplyImmediately=true` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.
- `ReplicationGroupId` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.

- `NodeGroupCount` – Obligatoire. Spécifie le nombre de partitions (groupes de nœuds) qui doivent exister une fois l'opération terminée. Lors de l'ajout des partitions, la valeur de `NodeGroupCount` doit être supérieure au nombre actuel de partitions.

  Le cas échéant, vous pouvez spécifier la zone de disponibilité de chaque nœud du groupe de réplication avec `ReshardingConfiguration`.
- `ReshardingConfiguration` : facultatif. Liste des zones de disponibilité favorites de chaque nœud de chaque partition du groupe de réplication. N'utilisez ce paramètre que si la valeur de `NodeGroupCount` est supérieure au nombre actuel de partitions. Si ce paramètre est omis lors de l'ajout de partitions, Amazon ElastiCache sélectionne les zones de disponibilité pour les nouveaux nœuds.

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en ajoutant des partitions à l'aide de l'API. ElastiCache

Example - Ajout de partitions

L'exemple suivant ajoute des groupes de nœuds au cluster Valkey ou Redis OSS (mode cluster activé)`my-cluster`, de sorte qu'il y a un total de quatre groupes de nœuds à la fin de l'opération. L'exemple spécifie aussi la zone de disponibilité de chaque nœud de chaque partition. L'opération commence immédiatement.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=4
    &ReplicationGroupId=my-cluster

 &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2a

 &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone
east-2c

 &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2b

 &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone
east-2a
```

```
 &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2c

 &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone
east-2d

 &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2d

 &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone
east-2c
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez ModifyReplicationGroupShardConfigurationla référence de ElastiCache l'API.

Suppression de partitions avec le repartitionnement en ligne

Vous pouvez supprimer des fragments de votre cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de l'API AWS Management Console AWS CLI, ou. ElastiCache

Rubriques

- Suppression de partitions (console)
- Suppression de partitions (AWS CLI)
- Supprimer des fragments (ElastiCacheAPI)

Suppression de partitions (console)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en supprimant les partitions à l'aide du. AWS Management Console

Avant de supprimer des groupes de nœuds (partitions) de votre groupe de réplication, ElastiCache assurez-vous que toutes vos données rentreront dans les partitions restantes. Si c'est le cas, les partitions spécifiées sont supprimées du groupe de réplication comme requis. Si les données ne tiennent pas dans les groupes de nœuds restants, la procédure est interrompue et le groupe de

réplication est maintenu dans la même configuration de groupe de nœuds qu'avant que la demande ne soit envoyée.

Vous pouvez utiliser le AWS Management Console pour supprimer une ou plusieurs partitions de votre cluster Valkey ou Redis OSS (mode cluster activé). Vous pouvez supprimer toutes les partitions d'un groupe de réplication. Vous devez à la place supprimer l'ancien groupe de réplication. Pour de plus amples informations, veuillez consulter Suppression d'un groupe de réplication. La procédure suivante décrit la suppression d'une ou plusieurs partitions.

Pour supprimer des fragments de votre cluster Valkey ou Redis OSS (mode cluster activé)

1. Ouvrez la ElastiCache console à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Recherchez et choisissez le nom, et non la case à gauche du nom du cluster, du cluster Valkey ou Redis OSS (mode cluster activé) dont vous souhaitez supprimer des partitions.

   > **Tip**
   >
   > Les clusters Valkey ou Redis OSS (mode cluster activé) ont une valeur supérieure ou égale à 1 dans la colonne Shards.

4. Dans la liste des partitions, cochez la case à gauche du nom de chaque partition que vous souhaitez supprimer.
5. Choisissez Supprimer une partition.

Suppression de partitions (AWS CLI)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en supprimant les partitions à l'aide du. AWS CLI

> **Important**
>
> Avant de supprimer des groupes de nœuds (partitions) de votre groupe de réplication, ElastiCache assurez-vous que toutes vos données rentreront dans les partitions restantes. Si c'est le cas, les partitions spécifiées (`--node-groups-to-remove`) sont supprimés du groupe de réplication comme demandé, puis leurs espaces de clés sont mappés dans les partitions restantes. Si les données ne tiennent pas dans les groupes de nœuds restants,

la procédure est interrompue et le groupe de réplication est maintenu dans la même configuration de groupe de nœuds qu'avant que la demande ne soit envoyée.

Vous pouvez utiliser le AWS CLI pour supprimer une ou plusieurs partitions de votre cluster Valkey ou Redis OSS (mode cluster activé). Vous pouvez supprimer toutes les partitions d'un groupe de réplication. Vous devez à la place supprimer l'ancien groupe de réplication. Pour de plus amples informations, veuillez consulter Suppression d'un groupe de réplication.

Utilisez les paramètres suivants avec `modify-replication-group-shard-configuration`.

Paramètres

- `--apply-immediately` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.
- `--replication-group-id` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.
- `--node-group-count` – Obligatoire. Spécifie le nombre de partitions (groupes de nœuds) qui doivent exister une fois l'opération terminée. Lors de la suppression des partitions, la valeur de `--node-group-count` doit être inférieure au nombre actuel de partitions.
- `--node-groups-to-remove` – Obligatoire quand `--node-group-count` est inférieur au nombre actuel de groupes de nœuds (partitions). Liste des partitions (groupe de nœuds) IDs à supprimer du groupe de réplication.

La procédure suivante décrit la suppression d'une ou plusieurs partitions.

Example – Suppression de partitions

L'exemple suivant supprime deux groupes de nœuds du cluster Valkey ou Redis OSS (mode cluster activé)`my-cluster`, de sorte qu'il y a un total de deux groupes de nœuds à la fin de l'opération. Les espaces de clés des partitions supprimées sont répartis également sur les partitions restantes.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 2 \
```

```
    --node-groups-to-remove "0002" "0003" \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group-shard-configuration ^
    --replication-group-id my-cluster ^
    --node-group-count 2 ^
    --node-groups-to-remove "0002" "0003" ^
    --apply-immediately
```

Supprimer des fragments (ElastiCacheAPI)

Vous pouvez utiliser l' ElastiCache API pour reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en ligne en utilisant l'opération. `ModifyReplicationGroupShardConfiguration`

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en supprimant les partitions à l'aide de l'API. ElastiCache

> **Important**
>
> Avant de supprimer des groupes de nœuds (partitions) de votre groupe de réplication, ElastiCache assurez-vous que toutes vos données rentreront dans les partitions restantes. Si c'est le cas, les partitions spécifiées (`NodeGroupsToRemove`) sont supprimés du groupe de réplication comme demandé, puis leurs espaces de clés sont mappés dans les partitions restantes. Si les données ne tiennent pas dans les groupes de nœuds restants, la procédure est interrompue et le groupe de réplication est maintenu dans la même configuration de groupe de nœuds qu'avant que la demande ne soit envoyée.

Vous pouvez utiliser l' ElastiCache API pour supprimer une ou plusieurs partitions de votre cluster Valkey ou Redis OSS (mode cluster activé). Vous pouvez supprimer toutes les partitions d'un groupe de réplication. Vous devez à la place supprimer l'ancien groupe de réplication. Pour de plus amples informations, veuillez consulter Suppression d'un groupe de réplication.

Utilisez les paramètres suivants avec `ModifyReplicationGroupShardConfiguration`.

Paramètres

- `ApplyImmediately=true` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.
- `ReplicationGroupId` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.
- `NodeGroupCount` – Obligatoire. Spécifie le nombre de partitions (groupes de nœuds) qui doivent exister une fois l'opération terminée. Lors de la suppression des partitions, la valeur de `NodeGroupCount` doit être inférieure au nombre actuel de partitions.
- `NodeGroupsToRemove` – Obligatoire quand `--node-group-count` est inférieur au nombre actuel de groupes de nœuds (partitions). Liste des partitions (groupe de nœuds) IDs à supprimer du groupe de réplication.

La procédure suivante décrit la suppression d'une ou plusieurs partitions.

Example – Suppression de partitions

L'exemple suivant supprime deux groupes de nœuds du cluster Valkey ou Redis OSS (mode cluster activé)`my-cluster`, de sorte qu'il y a un total de deux groupes de nœuds à la fin de l'opération. Les espaces de clés des partitions supprimées sont répartis également sur les partitions restantes.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=2
    &ReplicationGroupId=my-cluster
    &NodeGroupsToRemove.member.1=0002
    &NodeGroupsToRemove.member.2=0003
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Rééquilibrage en ligne des partitions

Vous pouvez rééquilibrer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) à l'aide de l'API AWS Management Console, AWS CLI ou. ElastiCache

Rubriques

- Rééquilibrage en ligne des partitions (console)
- Rééquilibrage des partitions en ligne (AWS CLI)
- Rééquilibrage des partitions en ligne (API) ElastiCache

Rééquilibrage en ligne des partitions (console)

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en rééquilibrant les partitions à l'aide du. AWS Management Console

Pour rééquilibrer les espaces clés entre les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé)

1. Ouvrez la ElastiCache console à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Choisissez le nom, et non la case à gauche du nom, du cluster Valkey ou Redis OSS (mode cluster activé) que vous souhaitez rééquilibrer.

   > **Tip**
   >
   > Les clusters Valkey ou Redis OSS (mode cluster activé) ont une valeur supérieure ou égale à 1 dans la colonne Shards.

4. Choisissez Rééquilibrer.
5. À l'invite, choisissez Rééquilibrer. Vous verrez peut-être un message similaire à celui-ci :*`Slots in the replication group are uniformly distributed. Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)`*. Sinon, choisissez Annuler.

Rééquilibrage des partitions en ligne (AWS CLI)

Utilisez les paramètres suivants avec `modify-replication-group-shard-configuration`.

Paramètres

- `-apply-immediately` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.

- `--replication-group-id` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.
- `--node-group-count` – Obligatoire. Pour rééquilibrer l'espace de clés entre toutes les partitions du cluster, cette valeur doit être identique au nombre actuel de partitions.

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en rééquilibrant les partitions à l'aide du. AWS CLI

Example – Rééquilibrage des partitions d'un cluster

L'exemple suivant rééquilibre les emplacements du cluster Valkey ou Redis OSS (mode cluster activé) `my-cluster` afin qu'ils soient répartis de la manière la plus égale possible. La valeur de `--node-group-count` (4) est le nombre de partitions actuellement dans le cluster.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id my-cluster \
    --node-group-count 4 \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group-shard-configuration ^
    --replication-group-id my-cluster ^
    --node-group-count 4 ^
    --apply-immediately
```

Rééquilibrage des partitions en ligne (API) ElastiCache

Vous pouvez utiliser l' ElastiCache API pour reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en ligne en utilisant l'opération. `ModifyReplicationGroupShardConfiguration`

Utilisez les paramètres suivants avec `ModifyReplicationGroupShardConfiguration`.

Paramètres

- `ApplyImmediately=true` – Obligatoire. Spécifie que l'opération de reconfiguration des partitions doit démarrer immédiatement.

- `ReplicationGroupId` – Obligatoire. Spécifie sur quel groupe de réplication (cluster) l'opération de reconfiguration des partitions doit être exécutée.
- `NodeGroupCount` – Obligatoire. Pour rééquilibrer l'espace de clés entre toutes les partitions du cluster, cette valeur doit être identique au nombre actuel de partitions.

Le processus suivant décrit comment reconfigurer les partitions de votre cluster Valkey ou Redis OSS (mode cluster activé) en rééquilibrant les partitions à l'aide de l'API. ElastiCache

Example – Rééquilibrage d'un cluster

L'exemple suivant rééquilibre les emplacements du cluster Valkey ou Redis OSS (mode cluster activé) `my-cluster` afin qu'ils soient répartis de la manière la plus égale possible. La valeur de `NodeGroupCount` (4) est le nombre de partitions actuellement dans le cluster.

```
https://elasticache.us-east-2.amazonaws.com/
    ?Action=ModifyReplicationGroupShardConfiguration
    &ApplyImmediately=true
    &NodeGroupCount=4
    &ReplicationGroupId=my-cluster
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20171002T192317Z
    &X-Amz-Credential=<credential>
```

Dimensionnement vertical en ligne en modifiant le type de nœud

En utilisant la mise à l'échelle verticale en ligne avec Valkey version 7.2 ou ultérieure, ou Redis OSS version 3.2.10 ou ultérieure, vous pouvez dimensionner vos clusters Valkey ou Redis OSS de manière dynamique avec un minimum de temps d'arrêt. Cela permet à votre cluster Valkey ou Redis OSS de répondre aux demandes même pendant le dimensionnement.

> **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 de plus amples informations, veuillez consulter Hiérarchisation des données ElastiCache.

Vous pouvez effectuer les actions suivantes :

- Élargissement : augmentez la capacité de lecture et d'écriture en ajustant le type de nœud de votre cluster Valkey ou Redis OSS pour utiliser un type de nœud plus grand.

  ElastiCache 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, redimensionne ElastiCache 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. Pour garantir un up/down flux d'échelle fluide, procédez comme suit :
>
> - Assurez-vous d'avoir une capacité ENI (interface réseau Elastic) suffisante. Pour une réduction, assurez-vous que la mémoire du plus petit nœud est suffisante pour absorber le trafic attendu.
>
> Pour accéder aux bonnes pratiques en matière de gestion de la mémoire, veuillez consulter Gestion de la mémoire réservée pour Valkey et Redis OSS.
> - 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.

Table des matières

- Augmentation en ligne
  - Mise à l'échelle des clusters de cache Valkey ou Redis OSS (console)
  - Mise à l'échelle des clusters de cache Valkey ou Redis OSS ()AWS CLI

- Mise à l'échelle des clusters de cache (ElastiCache API) Valkey ou Redis OSS
- Réduction en ligne
  - Réduction de la taille des clusters de cache Valkey ou Redis OSS (console)
  - Réduction de la taille des clusters de cache Valkey ou Redis OSS ()AWS CLI
  - Réduction de la taille des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Augmentation en ligne

Rubriques

- Mise à l'échelle des clusters de cache Valkey ou Redis OSS (console)
- Mise à l'échelle des clusters de cache Valkey ou Redis OSS ()AWS CLI
- Mise à l'échelle des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Mise à l'échelle des clusters de cache Valkey ou Redis OSS (console)

La procédure suivante décrit comment augmenter la taille d'un cluster Valkey ou Redis OSS à l'aide de la console de ElastiCache gestion. Au cours de ce processus, votre cluster continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour agrandir un cluster Valkey ou Redis OSS (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Choisissez le cluster dans la liste.
4. Sélectionnez Modifier.
5. Dans l'assistant 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.
6. Si vous souhaitez effectuer un processus d'augmentation immédiatement, cochez la case Apply immediately (Appliquer immédiatement). Si la case Appliquer immédiatement est décochée, le processus de mise à l'échelle est effectué lors du créneau de maintenance suivant du cluster.
7. Sélectionnez Modifier.

Si vous avez choisi Apply immediately lors de l'étape précédente, le statut du cluster passe à 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 de cache Valkey ou Redis OSS ()AWS CLI

La procédure suivante décrit comment augmenter la taille d'un cluster de cache Valkey ou Redis OSS à l'aide du. AWS CLI Au cours de ce processus, votre cluster continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour augmenter la taille d'un cluster de cache Valkey ou Redis OSS ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez augmenter en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache list-allowed-node-type-modifications \
        --replication-group-id my-replication-group-id
   ```

   Pour Windows :

   ```
   aws elasticache list-allowed-node-type-modifications ^
        --replication-group-id my-replication-group-id
   ```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

   ```
   {
        "ScaleUpModifications": [
            "cache.m3.2xlarge",
            "cache.m3.large",
            "cache.m3.xlarge",
            "cache.m4.10xlarge",
            "cache.m4.2xlarge",
            "cache.m4.4xlarge",
            "cache.m4.large",
            "cache.m4.xlarge",
            "cache.r3.2xlarge",
            "cache.r3.4xlarge",
   ```

```
        "cache.r3.8xlarge",
        "cache.r3.large",
        "cache.r3.xlarge"
    ]
       "ScaleDownModifications": [
        "cache.t2.micro",
        "cache.t2.small ",
        "cache.t2.medium",
        "cache.t1.small "
    ],
}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Modifiez votre groupe de réplication pour l'adapter au nouveau type de nœud plus grand à l'aide de la AWS CLI `modify-replication-group` commande et des paramètres suivants.

   - `--replication-group-id` : le nom du groupe de réplication que vous augmentez.
   - `--cache-node-type` : le nouveau type de nœud pour lequel vous voulez mettre à l'échelle le cluster de cache. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-modifications` lors de l'étape 1.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` : ce paramètre entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant du cluster, utilisez le paramètre `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group  \
     --replication-group-id my-redis-cluster \
     --cache-node-type cache.m3.xlarge \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-replication-group ^
     --replication-group-id my-redis-cluster ^
     --cache-node-type cache.m3.xlarge ^
     --apply-immediately
```

Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
  "ReplicationGroup": {
        "Status": "modifying",
        "Description": "my-redis-cluster",
        "NodeGroups": [
            {
                "Status": "modifying",
                "Slots": "0-16383",
                "NodeGroupId": "0001",
                "NodeGroupMembers": [
                    {
                        "PreferredAvailabilityZone": "us-east-1f",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-001"
                    },
                    {
                        "PreferredAvailabilityZone": "us-east-1d",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-002"
                    }
                ]
            }
        ],
        "ConfigurationEndpoint": {
            "Port": 6379,
            "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
        },
        "ClusterEnabled": true,
        "ReplicationGroupId": "my-redis-cluster",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "07:30-08:30",
        "MemberClusters": [
```

```
            "my-redis-cluster-0001-001",
            "my-redis-cluster-0001-002"
        ],
        "CacheNodeType": "cache.m3.xlarge",
         "DataTiering": "disabled"
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations, consultez modify-replication-group dans la référence AWS CLI .

3. Si vous avez utilisé le`--apply-immediately`, vérifiez l'état du cluster de cache à l'aide de la AWS CLI `describe-cache-clusters` commande avec le paramètre suivant. Lorsque le statut passe à available (disponible), vous pouvez commencer à utiliser le nouveau nœud de cluster de cache plus grand.

Mise à l'échelle des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Le processus suivant fait passer votre cluster de cache de son type de nœud actuel à un nouveau type de nœud plus grand à l'aide de l' ElastiCache API. Au cours de ce processus, ElastiCache met à jour les entrées DNS afin qu'elles pointent vers les nouveaux nœuds. Ainsi, vous n'aurez plus besoin de mettre à jour les points de terminaison de votre application. Pour Valkey 7.2 et versions ultérieures, Redis OSS 5.0.5 et versions ultérieures, vous pouvez dimensionner les clusters compatibles avec le basculement automatique pendant que le cluster reste en ligne et traite les demandes entrantes. Sur les versions Redis OSS 4.0.10 et antérieures, vous remarquerez peut-être une brève interruption des lectures et des écritures sur les versions précédentes depuis le nœud principal pendant la mise à jour de l'entrée DNS.

La durée nécessaire pour remettre à l'échelle la taille d'un type de nœud et passer à un type plus grand, varie selon le type de nœud et la quantité de données dans votre cluster de actuel.

Pour agrandir un cluster de cache (ElastiCache API) Valkey ou Redis OSS

1. Déterminez les types de nœuds que vous pouvez augmenter à l'aide de l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `ReplicationGroupId` : le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication spécifique plutôt que tous les groupes de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ListAllowedNodeTypeModifications
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Adaptez votre groupe de réplication actuel au nouveau type de nœud à l'aide de l'action `ModifyReplicationGroup` ElastiCache API et avec les paramètres suivants.

   - `ReplicationGroupId` : le nom du groupe de réplication.
   - `CacheNodeType` : le nouveau type de nœud plus grand des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.
   - `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `ApplyImmediately` : lorsqu'il est défini sur `true`, il entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de mise à l'échelle au créneau de maintenance suivant, utilisez `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.m3.2xlarge
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
```

```
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez ModifyReplicationGroup le Amazon ElastiCache API Reference.

3. Si vous l'avez utilisé `ApplyImmediately=true`, surveillez l'état du groupe de réplication à l'aide de l'`DescribeReplicationGroups`action ElastiCache API avec les paramètres suivants. Lorsque le statut passe de modifying à available, cela signifie que vous pouvez commencer à écrire sur votre nouveau groupe de réplication redimensionné.

   - `ReplicationGroupId` – Le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication particulier plutôt que tous les groupes de réplication.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeReplicationGroups
    &ReplicationGroupId=MyReplGroup
    &Version=2015-02-02
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20150202T192317Z
    &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez DescribeReplicationGroups le Amazon ElastiCache API Reference.

Réduction en ligne

Rubriques

- Réduction de la taille des clusters de cache Valkey ou Redis OSS (console)
- Réduction de la taille des clusters de cache Valkey ou Redis OSS ()AWS CLI
- Réduction de la taille des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Réduction de la taille des clusters de cache Valkey ou Redis OSS (console)

La procédure suivante décrit comment réduire la taille d'un cluster Valkey ou Redis OSS à l'aide de la console de ElastiCache gestion. Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour réduire la taille d'un cluster Valkey ou Redis OSS (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Choisissez votre cluster préféré dans la liste.
4. Sélectionnez Modifier.
5. Dans l'assistant 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.
6. Si vous souhaitez effectuer un processus de réduction immédiatement, cochez la case Apply immediately (Appliquer immédiatement). Si la case Apply immediately (Appliquer immédiatement) est décochée, le processus de réduction sera effectué lors de la prochaine fenêtre de maintenance du cluster.
7. Sélectionnez Modifier.

   Si vous avez choisi Apply immediately lors de l'étape précédente, le statut du cluster passe à 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 de cache Valkey ou Redis OSS ()AWS CLI

La procédure suivante décrit comment réduire la taille d'un cluster de cache Valkey ou Redis OSS à l'aide du. AWS CLI Au cours de ce processus, votre cluster continuera à traiter les demandes avec un temps d'arrêt minimal.

Pour réduire la taille d'un cluster de cache Valkey ou Redis OSS ()AWS CLI

1. Déterminez les types de nœuds que vous pouvez réduire en exécutant la AWS CLI `list-allowed-node-type-modifications` commande avec le paramètre suivant.

Pour Linux, macOS ou Unix :

```
aws elasticache list-allowed-node-type-modifications \
     --replication-group-id my-replication-group-id
```

Pour Windows :

```
aws elasticache list-allowed-node-type-modifications ^
     --replication-group-id my-replication-group-id
```

Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
     "ScaleUpModifications": [
         "cache.m3.2xlarge",
         "cache.m3.large",
         "cache.m3.xlarge",
         "cache.m4.10xlarge",
         "cache.m4.2xlarge",
         "cache.m4.4xlarge",
         "cache.m4.large",
         "cache.m4.xlarge",
         "cache.r3.2xlarge",
         "cache.r3.4xlarge",
         "cache.r3.8xlarge",
         "cache.r3.large",
         "cache.r3.xlarge"
     ]

        "ScaleDownModifications": [
         "cache.t2.micro",
         "cache.t2.small ",
         "cache.t2.medium ",
         "cache.t1.small"
     ]
}
```

Pour plus d'informations, consultez list-allowed-node-type-modifications dans la référence AWS CLI .

2. Modifiez votre groupe de réplication pour le réduire au nouveau type de nœud plus petit, à l'aide de la AWS CLI `modify-replication-group` commande et des paramètres suivants.

   - `--replication-group-id` – Le nom du groupe de réplication que vous réduisez.
   - `--cache-node-type` – Le nouveau type de nœud pour lequel vous voulez mettre à l'échelle le cluster de cache. Cette valeur doit correspondre à l'un des types de nœuds renvoyés par la commande `list-allowed-node-type-modifications` lors de l'étape 1.
   - `--cache-parameter-group-name` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
   - `--apply-immediately` – Ce paramètre entraîne l'application immédiate du processus d'augmentation. Pour reporter le processus de réduction à la fenêtre de maintenance suivante du cluster, utilisez le paramètre `--no-apply-immediately`.

   Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group  \
     --replication-group-id my-redis-cluster \
     --cache-node-type cache.t2.micro \
     --apply-immediately
```

   Pour Windows :

```
aws elasticache modify-replication-group ^
     --replication-group-id my-redis-cluster ^
     --cache-node-type cache.t2.micro ^
     --apply-immediately
```

   Le résultat de la commande ci-dessus doit être similaire à ce qui suit (format JSON).

```
{
  "ReplicationGroup": {
        "Status": "modifying",
        "Description": "my-redis-cluster",
        "NodeGroups": [
```

```
            {
                "Status": "modifying",
                "Slots": "0-16383",
                "NodeGroupId": "0001",
                "NodeGroupMembers": [
                    {
                        "PreferredAvailabilityZone": "us-east-1f",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-001"
                    },
                    {
                        "PreferredAvailabilityZone": "us-east-1d",
                        "CacheNodeId": "0001",
                        "CacheClusterId": "my-redis-cluster-0001-002"
                    }
                ]
            }
        ],
        "ConfigurationEndpoint": {
            "Port": 6379,
            "Address": "my-redis-
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"
        },
        "ClusterEnabled": true,
        "ReplicationGroupId": "my-redis-cluster",
        "SnapshotRetentionLimit": 1,
        "AutomaticFailover": "enabled",
        "SnapshotWindow": "07:30-08:30",
        "MemberClusters": [
            "my-redis-cluster-0001-001",
            "my-redis-cluster-0001-002"
        ],
        "CacheNodeType": "cache.t2.micro",
         "DataTiering": "disabled"
        "PendingModifiedValues": {}
    }
}
```

Pour plus d'informations, consultez modify-replication-group dans la référence AWS CLI .

3. Si vous avez utilisé le`--apply-immediately`, vérifiez l'état du cluster de cache à l'aide de la AWS CLI `describe-cache-clusters` commande avec le paramètre suivant. Lorsque le

statut passe à available (disponible), vous pouvez commencer à utiliser le nouveau nœud du cluster de cache plus petit.

Réduction de la taille des clusters de cache (ElastiCache API) Valkey ou Redis OSS

Le processus suivant fait passer votre groupe de réplication de son type de nœud actuel à un nouveau type de nœud plus petit à l'aide de l' ElastiCache API. Au cours de ce processus, votre cluster Valkey ou Redis OSS continuera à traiter les demandes avec un temps d'arrêt minimal.

La durée nécessaire à la réduction vers un type de nœud plus petit varie selon le type de nœud et la quantité de données dans votre cluster de cache actuel.

Diminution (ElastiCache API)

1. Déterminez les types de nœuds que vous pouvez réduire à l'aide de l'`ListAllowedNodeTypeModifications`action d' ElastiCache API avec le paramètre suivant.

   - `ReplicationGroupId` : le nom du groupe de réplication. Utilisez ce paramètre pour décrire un groupe de réplication spécifique plutôt que tous les groupes de réplication.

   ```
   https://elasticache.us-west-2.amazonaws.com/
       ?Action=ListAllowedNodeTypeModifications
       &ReplicationGroupId=MyReplGroup
       &Version=2015-02-02
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20150202T192317Z
       &X-Amz-Credential=<credential>
   ```

   Pour plus d'informations, consultez ListAllowedNodeTypeModifications le Amazon ElastiCache API Reference.

2. Réduisez votre groupe de réplication actuel au nouveau type de nœud à l'aide de l'action d'`ModifyReplicationGroup` ElastiCache API et avec les paramètres suivants.

   - `ReplicationGroupId` : le nom du groupe de réplication.
   - `CacheNodeType` : le nouveau type de nœud plus petit des clusters de cache dans ce groupe de réplication. Cette valeur doit être l'un des types d'instance renvoyés par l'`ListAllowedNodeTypeModifications`action de l'étape précédente.

- `CacheParameterGroupName` : [Facultatif] Utilisez ce paramètre si vous avez recours à `reserved-memory` pour gérer la mémoire réservée de votre cluster. Spécifiez un groupe de paramètres de cache personnalisé qui réserve la quantité de mémoire nécessaire à votre nouveau type de nœud. Si vous utilisez `reserved-memory-percent`, vous pouvez omettre ce paramètre.
- `ApplyImmediately` : lorsqu'il est défini sur `true`, il entraîne l'application immédiate du processus de réduction. Pour reporter le processus de réduction à la prochaine fenêtre de maintenance, utilisez `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyReplicationGroup
    &ApplyImmediately=true
    &CacheNodeType=cache.t2.micro
    &CacheParameterGroupName=redis32-m3-2xl
    &ReplicationGroupId=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour plus d'informations, consultez ModifyReplicationGroup le Amazon ElastiCache API Reference.

Commencer à utiliser les filtres Bloom

ElastiCache prend en charge la structure de données du filtre Bloom, qui fournit une structure de données probabiliste peu encombrante pour vérifier si un élément fait partie d'un ensemble. Lorsque vous utilisez des filtres Bloom, des faux positifs sont possibles : un filtre peut indiquer de manière incorrecte l'existence d'un élément, même si cet élément n'a pas été ajouté à l'ensemble. Cependant, l'utilisation de filtres Bloom empêchera les faux négatifs, c'est-à-dire des indications incorrectes indiquant qu'un élément n'existe pas, même si cet élément a été ajouté à l'ensemble.

Vous pouvez définir le pourcentage de faux positifs potentiels à un taux adapté à votre charge de travail, en ajustant le taux fp. Vous pouvez également configurer la capacité (le nombre d'éléments qu'un filtre Bloom peut contenir), les propriétés dimensionnelles et non dimensionnelles, etc.

Une fois que vous avez créé un cluster avec une version de moteur prise en charge, le type de données Bloom et les commandes associées sont automatiquement disponibles. Le type de `bloom` données est compatible avec l'API avec la syntaxe de commande du filtre Bloom des bibliothèques clientes officielles de Valkey`valkey-py`, notamment`valkey-java`, et`valkey-go`. Vous pouvez facilement migrer les applications Valkey et Redis OSS existantes basées sur Bloom vers. ElastiCache Pour une liste complète des commandes, voirCommandes du filtre Bloom.

Les métriques liées à Bloom `BloomFilterBasedCmds``BloomFilterBasedCmdsLatency`, et `BloomFilterBasedCmdsECPUs` sont intégrées CloudWatch pour surveiller l'utilisation de ce type de données. Pour de plus amples informations, veuillez consulter Métriques pour Valkey et Redis OSS.

> **Note**
>
> - Pour utiliser les filtres Bloom, vous devez utiliser ElastiCache Valkey 8.1 ou version ultérieure.
> - Le type de données Bloom n'est pas compatible avec RDB avec d'autres offres Bloom non basées sur Valkey.

Vue d'ensemble des types de données des filtres Bloom

Les filtres Bloom sont une structure de données probabiliste peu encombrante qui permet d'ajouter des éléments et de vérifier si des éléments existent. Les faux positifs sont possibles lorsqu'un filtre indique de manière incorrecte l'existence d'un élément, même s'il n'a pas été ajouté. Cependant, les filtres Bloom garantissent l'absence de faux négatifs (indiquant à tort qu'un élément n'existe pas, même s'il a été ajouté).

La principale source de documentation pour les filtres Bloom se trouve sur la page de documentation de valkey.io. Il contient les informations suivantes :

- Cas d'utilisation courants des filtres Bloom
  - Publicité/Déduplication d'événements
  - Détection des fraudes

- Filtrer les contenus dommageables et les spams
- Détection unique des utilisateurs
- Différences entre les filtres de floraison scalables et non scalables
  - Comment choisir entre des filtres de floraison redimensionnables et non scalables
- Propriétés de la floraison
  - Découvrez les propriétés réglables des filtres Bloom. Cela inclut le taux de faux positifs, la capacité, les propriétés de mise à l'échelle et de non-mise à l'échelle, etc.
- Exécution des commandes Bloom
- Surveillance des statistiques générales du filtre Bloom
- Manipulation de grands filtres Bloom
  - Recommandations et informations sur la manière de vérifier si un filtre Bloom atteint sa limite d'utilisation de mémoire et s'il peut être redimensionné pour atteindre la capacité souhaitée.
  - Vous pouvez vérifier spécifiquement la quantité de mémoire consommée par un document de filtre Bloom à l'aide de la commande BF.INFO.

Limite de taille des fleurs

La consommation de mémoire par un seul objet du filtre Bloom est limitée à 128 Mo. Vous pouvez vérifier la quantité de mémoire consommée par un filtre Bloom à l'aide de la `BF.INFO <key> SIZE` commande.

Bloom ACLs

À l'instar des catégories existantes par type de données (@string, @hash, etc.), une nouvelle catégorie @bloom est ajoutée pour simplifier la gestion de l'accès aux commandes et aux données Bloom. Aucune autre commande Valkey ou Redis OSS existante n'appartient à la catégorie @bloom.

Trois catégories d'ACL existantes ont été mises à jour pour inclure les nouvelles commandes Bloom : @read, @write et @fast. Le tableau suivant indique le mappage des commandes Bloom aux catégories appropriées.

commande Bloom	@read	@write	@fast	@bloom
BF.AJOUTER		y	y	y

commande Bloom	@read	@write	@fast	@bloom
BF.CARD	y		y	y
BF.EXISTE	y		y	y
BF.INFO	y		y	y
BF.INSERT		y	y	y
BF.MADD		y	y	y
BF.MEXIST	y		y	y
BF.RESERVE	y		y	y

Métriques liées au filtre Bloom

Les CloudWatch métriques suivantes relatives aux structures de données Bloom sont fournies :

Métriques CW	Unit	Sans serveur/conçu par vos soins	Description
BloomFilterBasedCmds	Nombre	les deux	Le nombre total de commandes du filtre Bloom, y compris les commandes de lecture et d'écriture.
BloomFilterBasedCmdsLatency	Microsecondes	Autogéré	Latence de toutes les commandes du filtre Bloom, y compris les commandes de lecture et d'écriture.
BloomFilterBasedCmdsECPUs	Nombre	sans serveur	ECPUs consommé par toutes les

Métriques CW	Unit	Sans serveur/conçu par vos soins	Description
			commandes du filtre Bloom, y compris les commandes de lecture et d'écriture.

Commandes du filtre Bloom

Les commandes du filtre Bloom sont documentées sur le site Web de Valkey.io. Chaque page de commande fournit un aperçu complet des commandes bloom, notamment de leur syntaxe, de leur comportement, de leurs valeurs renvoyées et des conditions d'erreur potentielles.

Name (Nom)	Description
BF.AJOUTER	Ajoute un seul élément à un filtre Bloom. Si le filtre n'existe pas déjà, il est créé.
CARTE BF	Renvoie la cardinalité d'un filtre Bloom.
BF. EXISTE	Détermine si le filtre Bloom contient l'élément spécifié.
BF.INFO	Renvoie les informations d'utilisation et les propriétés d'un filtre Bloom spécifique.
BF.INSÉRER	Crée un filtre de floraison avec 0 éléments ou plus, ou ajoute des éléments à un filtre de floraison existant.
B.F. MADD	Ajoute un ou plusieurs éléments à un filtre Bloom.
B.F. EXISTE	Détermine si le filtre Bloom contient un ou plusieurs éléments.

Name (Nom)	Description
B.RESERVE	Crée un filtre de floraison vide avec les propriétés spécifiées.

> **Note**
>
> BF.LOAD n'est pas pris en charge par. ElastiCache Cela ne concerne que l'utilisation de l'AOF, qui n'est ElastiCache pas prise en charge.

Commencer à utiliser JSON pour Valkey et Redis OSS

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

En plus d'utiliser les opérations d'API Valkey et 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. Cela 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. Compatible avec les API et compatible RDB avec la version 2 du module JSON, vous pouvez donc facilement migrer les applications Valkey et Redis OSS existantes basées sur JSON vers. ElastiCache Pour plus d'informations sur les commandes prises en charge, consultezCommandes Valkey et Redis OSS prises en charge.

Les métriques `JsonBasedCmds` liées au JSON `JsonBasedCmdsLatency` sont intégrées CloudWatch pour surveiller l'utilisation de ce type de données. Pour plus d'informations, consultez Metrics for Valkey et Redis OSS.

> **Note**
>
> Pour utiliser JSON, vous devez exécuter Valkey 7.2 ou version ultérieure, ou Redis OSS 6.2.6 ou version ultérieure.

Rubriques

- Vue d'ensemble des types de données JSON
- Commandes Valkey et Redis OSS prises en charge

Vue d'ensemble des types de données JSON

ElastiCache prend en charge un certain nombre de commandes Valkey et 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 prises en charge.

Terminologie

Terme	Description
Document JSON	Fait référence à la valeur d'une clé JSON.
Valeur JSON	Fait référence à un sous-ensemble d'un document JSON, y compris la racine qui représente le document entier. Une valeur peut être un conteneur ou une entrée dans un conteneur.
Élément JSON	Équivalent de la 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 entraîne généralement une consommation de mémoire un peu plus importante 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 de la chaîne JSON. Vous pouvez vérifier la quantité de mémoire consommée par un document JSON en utilisant la commande `JSON.DEBUG MEMORY`.

JSON ACLs

- Similaire aux 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 Valkey ou 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 Valkey et Redis OSS existantes ont été mises à jour pour inclure les nouvelles commandes JSON : @read, @write, @fast, @slow et @admin. Le tableau suivant indique le mappage des commandes JSON aux catégories appropriées.

ACL

Commande JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		

Commande JSON	@read	@write	@fast	@slow	@admin
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		
JSON.NUMMULTBY		y	y		
JSON.OBJKEYS	y		y		

Commande JSON	@read	@write	@fast	@slow	@admin
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 comme premier argument. Certaines commandes ont également un argument path. L'argument path correspond par défaut à la racine s'il est optionnel et non fourni.

Notation :

- Les arguments obligatoires sont entourés de chevrons. Par exemple : <key>
- Les arguments facultatifs sont entourés de crochets. Par exemple : [path]
- Les arguments facultatifs supplémentaires sont indiqués par une ellipse (« … »). Par exemple : [json ...]

Syntaxe de chemin

Le JSON de Redis prend en charge deux types de syntaxe de chemin :

- Syntaxe améliorée — Suit la JSONPath syntaxe décrite par Goessner, comme indiqué dans le tableau suivant. 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 sont sensibles au 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 []	Opérateur enfant.
..	Descente récursive.
*	Caractère générique. Tous les éléments d'un objet ou d'un tableau.

Symbole/Expression	Description
[]	Opérateur d'indice de tableau. L'index est basé sur 0.
[,]	Opérateur d'union.
[start:end:step]	Opérateur de découpage de tableau.
?()	Applique une expression de filtre (script) au tableau ou à l'objet en cours.
()	Expression de filtre.
@	Utilisé dans les expressions de filtre qui font référence au nœud en cours de traitement.
==	Égal à, utilisé dans les expressions de filtre.
!=	Pas égal à, 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.
&&	AND logique, utilisé pour combiner plusieurs expressions de filtre.
||	OR logique, utilisé pour combiner plusieurs expressions de filtre.

Exemples

Les exemples suivants sont basés sur l'exemple de données XML de Goessner, que nous avons modifié 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 la boutique.
$..*	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]	Les livres d'index 0 à 3 (l'index final n'est pas inclusif).
$..book[0:4:2]	Les livres à l'index 0, 2.
$..book[?(@.isbn)]	Tous les livres ayant un numéro ISBN.
$..book[?(@.price<10)]	Tous les livres dont le prix est inférieur à 10 USD.
'$..book[?(@.price < 10)]'	Tous les livres dont le prix est inférieur à 10 USD. (Le chemin doit être entre guillemets s'il contient des espaces blancs).

Chemin	Description
'$..book[?(@["price"] < 10)]'	Tous les livres dont le prix est inférieur à 10 USD.
'$..book[?(@.["price"] < 10)]'	Tous les livres dont le prix est inférieur à 10 USD.
$.. book [? (@.prix>=10&&@.prix<=100)]	Tous les livres dans la fourchette de prix comprise entre 10 et 100 USD.
'$..book[?(@.price>=10 && @.price<=100)]'	Tous les livres dans la fourchette de prix comprise entre 10 et 100 USD. (Le chemin doit être entre guillemets s'il contient des espaces blancs).
$..book[?(@.sold==true||@.in-stock==false)]	Tous les livres vendus ou en rupture de stock.
'$..book[?(@.sold == true || @.in-stock == false)]'	Tous les livres vendus ou en rupture de stock. (Le chemin doit être entre guillemets s'il contient des espaces blancs).
'$.store.book[?(@.["category"] == "fiction")]'	Tous les livres dans la catégorie fiction.
'$.store.book[?(@.["category"] != "fiction")]'	Tous les livres dans la catégorie non-fiction.

Exemples d'expressions de filtre supplémentaires :

```
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 []	Opérateur enfant.
[]	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	Le nom de la ville.
["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	Une erreur qui se produit lorsque la limite de taille est dépassée. Par exemple, la limite de taille du document ou la limite de profondeur d'imbrication a été dépassée.
NONEXISTENT	Une clé ou un chemin n'existe pas.
OUTOFBOUNDARIES	L'index du tableau est hors limites.
SYNTAXERR	Erreur de syntaxe.
WRONGTYPE	Mauvais type de valeur.

Métriques liées à 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 Valkey ou Redis OSS.

Pour interroger les métriques principales, exécutez la commande suivante :

```
info json_core_metrics
```

Comment ElastiCache pour Valkey et Redis OSS interagit avec JSON

La section suivante décrit comment ElastiCache Valkey et Redis OSS 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 à l'intérieur des parenthèses sont exécutées en premier.

Comportement de la limite maximale d'imbrication des chemins

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

Traitement des valeurs numériques

JSON n'a pas de types de données séparés pour les nombres entiers et les nombres à virgule flottante. Ils sont tous appelés des nombres.

Représentations numériques :

Lorsqu'un nombre JSON est reçu en entrée, il est converti en l'une des deux représentations binaires internes : un entier signé de 64 bits ou une 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.

Commandes arithmétiques `NUMINCRBY` et `NUMMULTBY` :

- 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 Valkey et Redis OSS prises en charge.

Filtrage direct de tableau

ElastiCache pour Valkey ou Redis OSS filtre directement les objets du tableau.

Pour des données comme `[0,1,2,3,4,5,6]` et une requête de chemin comme`$[?(@<4)]`, ou des données similaires `{"my_key":[0,1,2,3,4,5,6]}` et une requête de chemin comme`$.my_key[?(@<4)]`, ElastiCache renverrait [1,2,3] dans les deux cas.

Comportement d'indexation de tableau

ElastiCache pour Valkey ou Redis OSS autorise les indices positifs et négatifs pour les tableaux. Pour un tableau de longueur cinq, 0 interrogerait le premier élément, 1 le deuxième, et ainsi de suite. Les nombres négatifs commencent à la fin du tableau, donc -1 interrogerait le cinquième élément, -2 le quatrième élément, et ainsi de suite.

Pour garantir un comportement prévisible aux clients, ElastiCache n'arrondit pas les index des tableaux vers le bas ou vers le haut. Ainsi, si vous avez un tableau d'une longueur de 5, le fait d'appeler l'index 5 ou plus, ou -6 ou moins, ne produira aucun résultat.

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 Valkey et Redis OSS prises en charge

ElastiCache prend en charge les commandes JSON Valkey et Redis OSS suivantes :

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.MSET
- JSON.NUMINCRBY
- JSON.NUMMULTBY
- JSON.OBJLEN
- JSON.OBJKEYS
- JSON.RESP
- JSON.SET
- JSON.STRAPPEND
- JSON.STRLEN
- JSON.TOGGLE
- JSON.TYPE

JSON.ARRAPPEND

Ajoute une ou plusieurs valeurs aux valeurs du tableau au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.

- json (obligatoire) : valeur JSON à ajouter au tableau.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représentent la nouvelle longueur du tableau au niveau de chaque chemin.
- Si une valeur n'est pas un tableau, sa valeur de retour correspondante est nulle.
- 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 premier 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

Recherche la première occurrence d'une valeur scalaire JSON dans les tableaux au niveau du 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) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.
- json-scalar (obligatoire) : valeur scalaire à rechercher. JSON scalar fait référence à des valeurs qui ne sont pas des objets ou des tableaux. C'est-à-dire que les chaînes, les nombres, les booléens et les valeurs·nulles sont des valeurs scalaires.
- start (facultatif) : index de départ, inclusif. La valeur par défaut est 0 si elle n'est pas fournie.
- end (facultatif) : index de fin, exclusif. La valeur par défaut est 0 si elle n'est pas fournie, 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

Insère une ou plusieurs valeurs dans le tableau des valeurs au niveau du chemin avant l'index.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.
- index (obligatoire) – Un index de 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 qui représentent la nouvelle longueur du tableau au niveau de 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

Obtient la longueur des valeurs du tableau au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représentent la longueur du tableau au niveau de 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 :

- 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 `NONEXISTENT JSON` 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"]]'
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 [1]
(integer) 1
127.0.0.1:6379> JSON.ARRLEN k1 [2]
(integer) 2

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]
(error) WRONGTYPE JSON element is not an array
127.0.0.1:6379> JSON.ARRLEN k2 [0]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 [6]
(error) OUTOFBOUNDARIES Array index is out of bounds
127.0.0.1:6379> JSON.ARRLEN k2 a.b
(error) NONEXISTENT JSON path does not exist
```

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) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

- index (facultatif) : La pposition dans le tableau à partir de laquelle l'extraction doit commencer.
  - 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 en bloc qui représentent les valeurs extraites au niveau de 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, qui représente la valeur JSON extraite.
- 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

Coupe un tableau au niveau du chemin pour qu'il devienne un sous tableau [start, end], les deux inclus.

- 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.ARRTRIM <key> <path> <start> <end>
```

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.
- start (obligatoire) – L'index de départ, inclusif.
- end (obligatoire) – L'index de fin, inclusif.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représentent la nouvelle longueur du tableau au niveau de 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

Efface les tableaux ou un objet au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

- Entier, le nombre de conteneurs effacés.
- L'effacement d'un tableau ou d'un objet vide compte pour 1 conteneur effacé.
- L'effacement d'une valeur non-conteneur retourne 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) 7
127.0.0.1:6379> JSON.CLEAR k1  $[*]
(integer) 4
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

Rapporte des informations. Les sous-commandes prises en charge sont :

- MEMORY <key> [path] – Rapporte 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.
- FIELDS <key> [path] – Rapporte le nombre de champs au niveau du 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.

- HELP – Imprime 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 qui représentent la taille de la mémoire (en octets) de la valeur JSON au niveau de chaque chemin.
  - Renvoie un tableau vide si la clé Valkey ou Redis OSS n'existe pas.
- Si le chemin est une syntaxe restreinte :
  - Renvoie un nombre entier, la taille de la mémoire et la valeur JSON en octets.
  - Renvoie null si la clé Valkey ou Redis OSS n'existe pas.

FIELDS

- Si le chemin est une syntaxe améliorée :
  - Renvoie un tableau d'entiers qui représentent le nombre de champs de la valeur JSON au niveau de chaque chemin.
  - Renvoie un tableau vide si la clé Valkey ou 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 Valkey ou 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

Supprime les valeurs JSON au niveau du chemin dans une clé de document. Si le chemin est la racine, cela revient à supprimer la clé de Valkey ou Redis OSS.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

- Nombre d'éléments supprimés.
- 0 si la clé Valkey ou 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) — Une clé Valkey ou 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 chacun d'elle est une chaîne vide. Elles peuvent être remplacées par n'importe quelle combinaison. Elles peuvent être spécifiées dans n'importe quel ordre.

- NOESCAPE : Facultatif, autorisé à être présent pour la compatibilité héritée et n'a aucun autre effet.
- path (facultatif) : zéro ou plusieurs chemins JSON, par défaut la racine si aucun n'est donné. 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 sérialisé, 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

Il est sérialisé JSONs sur le chemin à partir de plusieurs clés de document. Elle renvoie une valeur nulle pour une clé ou un chemin JSON inexistant.

Syntaxe

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

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

Retour

- Tableau de chaînes en bloc. La taille du tableau est égale au nombre de clés dans la commande. Chaque élément du tableau est complété par (a) le JSON sérialisé tel que localisé par le chemin

ou (b) une valeur nulle si la clé n'existe pas, le chemin n'existe pas dans le document, ou le chemin est invalide (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.MSET

Pris en charge pour Valkey version 8.1 et versions ultérieures.

Définissez des valeurs JSON pour plusieurs clés. L'opération est atomique. Toutes les valeurs sont définies ou aucune n'est définie.

Syntaxe

```
JSON.MSET key path json [ key path json ... ]
```

- 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 qu'il ne s'agit pas d'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.

Retour

- Réponse sous forme de chaîne simple : « OK » si l'opération a réussi.
- Réponse d'erreur simple : si l'opération a échoué.

Exemples

Syntaxe de chemin améliorée :

```
127.0.0.1:6379> JSON.MSET k1 . '[1,2,3,4,5]' k2 . '{"a":{"a":1, "b":2, "c":3}}' k3 .
 '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.GET k1
"[1,2,3,4,5]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.MSET k2 $.a.* '0' k3 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"
127.0.0.1:6379> JSON.GET k3
"{\"a\":[0,0,0,0,0]}"
```

Syntaxe de chemin restreinte :

```
127.0.0.1:6379> JSON.MSET k1 . '{"name": "John","address": {"street": "123 Main
 St","city": "Springfield"},"phones": ["555-1234","555-5678"]}'
OK
127.0.0.1:6379> JSON.MSET k1 .address.street '"21 2nd Street"' k1 .address.city '"New
 York"'
OK
127.0.0.1:6379> JSON.GET k1 .address.street
"\"21 2nd Street\""
127.0.0.1:6379> JSON.GET k1 .address.city
"\"New York\""
```

JSON.NUMINCRBY

Incrémente les valeurs numériques au niveau du chemin d'un nombre donné.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.
- number (obligatoire) : nombre.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes en bloc qui représente la valeur résultante au niveau de 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 qui représente 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

Multiplie les valeurs numériques au niveau du chemin par un nombre donné.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON.
- number (obligatoire) : nombre.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes en bloc qui représentent la valeur résultante au niveau de 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 d'un nombre à virgule flottante IEEE à double précision de 64 bits.
- `NONEXISTENT` si la clé du document n'existe pas.

Si le chemin est une syntaxe restreinte :

- Chaîne en bloc qui représente 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 en dehors de la plage d'un double IEEE de 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

Obtient le nombre de clés dans les valeurs de l'objet au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représentent la longueur de l'objet au niveau de 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 `NONEXISTENT JSON` 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

Obtient les noms des clés dans les valeurs de l'objet au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de 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.
- 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 Valkey ou 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 de JSON sont mappées aux chaînes simples respectives de RESP.
- 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 bloc de RESP.
- Les tableaux JSON sont représentés comme des tableaux RESP, où le premier élément est la chaîne simple [, suivie des éléments du tableau.
- Les objets JSON sont représentés comme des tableaux RESP, où le premier élément est la simple chaîne {, suivie des paires clé-valeur, chacune d'entre elles étant une chaîne en bloc de RESP.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de 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 qui représente la forme RESP de la valeur au niveau du 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

Définit les valeurs JSON au niveau du chemin.

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

- Si l'élément parent n'existe pas, la commande renvoie une erreur NONEXISTENT.
- Si l'élément parent existe, mais n'est pas un objet, la commande renvoie une erreur 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 renvoie une erreur NONEXISTENT.
  - 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 NONEXISTENT.
- 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 une 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) — Une clé Valkey ou Redis OSS de type document JSON.
- path (obligatoire) : chemin JSON. Pour une nouvelle clé, le chemin JSON doit être la racine «. ».
- NX (facultatif) — Si le chemin est la racine, définissez la valeur uniquement si la clé n'existe pas. C'est-à-dire, insérer un nouveau document. Si le chemin n'est pas la racine, définir la valeur uniquement si le chemin n'existe pas. C'est-à-dire, insérer une valeur dans le document.

- XX (facultatif) — Si le chemin est la racine, définissez la valeur uniquement si la clé existe. C'est-à-dire, remplacer le document existant. Si le chemin n'est pas la racine, définir la valeur uniquement si le chemin existe. C'est-à-dire, mettre 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

Ajoute une chaîne aux chaînes JSON au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.
- json_string (obligatoire) – La représentation JSON d'une chaîne. Notez qu'une chaîne JSON doit être entre guillemets. Par exemple : « "exemple de chaîne" ».

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représentent la nouvelle longueur de la chaîne au niveau de chaque chemin.
- Si une valeur au niveau du chemin n'est pas une chaîne, sa valeur de retour correspondante est nulle.
- Erreur `SYNTAXERR` si l'argument json en entrée 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 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

Obtient les longueurs des valeurs de chaîne JSON au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau d'entiers qui représente la longueur de la valeur de la chaîne au niveau de 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

Fait basculer les valeurs booléennes entre vrai et faux au niveau du chemin.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

Si le chemin est une syntaxe améliorée :

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

Si le chemin est une syntaxe restreinte :

- Chaîne (« vrai »/« faux ») qui représente la valeur booléenne résultante.
- `NONEXISTENT` si la clé du document n'existe pas.
- Erreur `WRONGTYPE` si la valeur au niveau 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

Rapporte le type de valeurs au niveau du chemin donné.

Syntaxe

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

- clé (obligatoire) — Une clé Valkey ou Redis OSS de type document JSON.
- path (facultatif) : chemin JSON. Par défaut, il s'agit de la racine si elle n'est pas fournie.

Retour

Si le chemin est une syntaxe améliorée :

- Tableau de chaînes de caractères qui représentent le type de valeur au niveau 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 ElastiCache

Pour vous aider à gérer vos clusters et autres ElastiCache ressources, 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 ElastiCache clusters de votre compte afin de suivre le propriétaire et le groupe d'utilisateurs de chaque instance.

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 ElastiCache et sont interprétées strictement comme des chaînes 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. En

outre, si vous ajoutez ou supprimez des balises sur un groupe de réplication, tous les nœuds de ce groupe de réplication verront également leurs balises ajoutées ou supprimées.

Vous pouvez travailler avec des balises à l'aide de l'API AWS Management Console AWS CLI, du et de l' ElastiCache API.

Si vous utilisez IAM, vous pouvez contrôler quels utilisateurs de votre AWS compte sont autorisés à créer, modifier ou supprimer des tags. Pour de plus amples informations, veuillez consulter Autorisations de niveau ressource.

Ressources que vous pouvez étiqueter

Vous pouvez étiqueter la plupart ElastiCache des ressources déjà présentes 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 ElastiCache service.

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' ElastiCache API Amazon, la AWS CLI ou un AWS SDK, vous pouvez utiliser le `Tags` paramètre de l'action d' ElastiCache API correspondante pour appliquer des balises. Il s'agit des options suivantes :

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`

- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

Le tableau suivant décrit les ElastiCache ressources qui peuvent être balisées et les ressources qui peuvent être balisées lors de la création à l'aide de l' ElastiCache API, de la AWS CLI ou d'un AWS SDK.

Support de balisage pour les ressources ElastiCache

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
Oui	Oui

Prend en charge les balises	Prend en charge le balisage au moment de la création
Oui	Oui
Oui	Oui
Oui	Oui
Oui	Oui

> **Note**
>
> Vous ne pouvez pas étiqueter les entrepôts de données globaux.

Vous pouvez appliquer des autorisations au niveau des ressources basées sur des balises dans vos politiques IAM aux actions d' ElastiCache API qui prennent en charge le balisage lors de la création afin de mettre en œuvre un contrôle granulaire sur les utilisateurs et les groupes qui peuvent 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 de plus amples informations, veuillez consulter 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.

Balisage des caches et des instantanés

Les règles suivantes s'appliquent à l'étiquetage dans le cadre d'opérations de requête :

- CreateReplicationGroup:

- Si les paramètres `--primary-cluster-id` et `--tags` sont inclus dans la requête, les balises de requête sont ajoutées au groupe de réplication et propagées à tous les clusters de cache du groupe de réplication. Si le cluster de cache principal possède des balises existantes, celles-ci seront remplacées par les balises de requête pour avoir des balises cohérentes sur tous les nœuds.

  S'il n'y a pas de balises de requête, les balises de cluster de cache primaires sont ajoutées au groupe de réplication et propagées à tous les clusters de cache.
- Si `--snapshot-name` ou `--serverless-cache-snapshot-name` est fourni :

  Si des balises sont incluses dans la demande, le groupe de réplication sera étiqueté uniquement avec ces balises. Si aucune balise n'est incluse dans la demande, les balises de l'instantané seront ajoutées au groupe de réplication.
- Si `--global-replication-group-id` est fourni :

  Si des balises sont incluses dans la demande, les balises de la demande seront ajoutées au groupe de réplication et propagées à tous les clusters de cache.

- CreateCacheCluster :
  - Si `--replication-group-id` est fourni :

    Si des balises sont incluses dans la requête, le cluster de cache sera étiqueté uniquement avec ces balises. Si aucune balise n'est incluse dans la demande, le cluster de cache héritera des balises de groupe de réplication au lieu des balises du cluster de cache principal.
  - Si `--snapshot-name` est fourni :

    Si des balises sont incluses dans la requête, le cluster de cache sera étiqueté uniquement avec ces balises. Si aucune balise n'est incluse dans la requête, les balises d'instantané seront ajoutées au cluster de cache.
- CreateServerlessCache :
  - Si des balises sont incluses dans la demande, seules les balises de la demande seront ajoutées au cache sans serveur.
- CreateSnapshot :
  - Si `--replication-group-id` 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 requête, les balises du groupe de réplication sont ajoutées à l'instantané.

- Si `--cache-cluster-id` 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 requête, les balises de cluster de cache sont ajoutées à l'instantané.

- Pour les instantanés automatiques :

  Les balises se propagent à partir des balises du groupe de réplication.

- CreateServerlessCacheSnapshot :
  - Si des balises sont incluses dans la demande, seules les balises de la demande seront ajoutées à l'instantané du cache sans serveur.
- 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.
- CopyServerlessCacheSnapshot :
  - Si des balises sont incluses dans la demande, seules les balises de la demande seront ajoutées à l'instantané du cache sans serveur.
- AddTagsToResourceet RemoveTagsFromResource:
  - Les balises added/removed proviendront du groupe de réplication et l'action sera propagée à tous les clusters du groupe de réplication.

> **Note**
>
> AddTagsToResourceet RemoveTagsFromResourcene peut pas être utilisé pour les paramètres par défaut et les groupes de sécurité.

- IncreaseReplicaCountet ModifyReplicationGroupShardConfiguration:
  - Tous les nouveaux clusters ajoutés au groupe de réplication auront les mêmes balises que le groupe de réplication.

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 qu'il ElastiCache 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 ElastiCache ressources que vous pouvez baliser, consultezRessources que vous pouvez étiqueter.

Exemple : étiquetage de vos ressources

- Création d'un cache sans serveur à l'aide de balises. Cet exemple utilise Memcached comme moteur.

```
aws elasticache create-serverless-cache \
    --serverless-cache-name CacheName \
    --engine memcached
    --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Ajout de balises à un cache sans serveur

```
aws elasticache add-tags-to-resource \
--resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Ajout de balises à un groupe de réplication.

```
aws elasticache add-tags-to-resource \
--resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Création d'un cluster de cache en utilisant des balises.

```
aws elasticache create-cache-cluster \
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Création d'un cluster de cache en utilisant des balises. Cet exemple utilise Redis comme moteur.

```
aws elasticache create-cache-cluster \
--cluster-id testing-tags \
--cluster-description cluster-test \
--cache-subnet-group-name test \
--cache-node-type cache.t2.micro \
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Création d'un instantané sans serveur avec des balises. Cet exemple utilise Memcached comme moteur.

```
aws elasticache create-serverless-cache-snapshot \
--serverless-cache-name testing-tags \
--serverless-cache-snapshot-name bkp-testing-tags-scs \
--tags Key="work",Value="foo"
```

- Création d'un instantané avec des balises.

Les instantanés ne sont actuellement disponibles que pour Redis. Dans ce cas, si vous ajoutez des balises sur demande, même si le groupe de réplication contient des balises, l'instantané recevra uniquement les balises demandées.

```
aws elasticache create-snapshot \
--replication-group-id testing-tags \
--snapshot-name bkp-testing-tags-rg \
--tags Key="work",Value="foo"
```

Exemples de politique de contrôle d'accès basée sur des balises

1. Autoriser une action `AddTagsToResource` sur un cluster uniquement si le cluster contient la balise Project=XYZ.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "elasticache:AddTagsToResource",
               "Resource": [
                   "arn:aws:elasticache:*:*:cluster:*"
               ],
               "Condition": {
                   "StringEquals": {
                       "aws:ResourceTag/Project": "XYZ"
                   }
               }
           }
       ]
   }
   ```

2. Autoriser une action `RemoveTagsFromResource` à partir d'un groupe de réplication s'il contient les balises Projet et Service et que les clés sont différentes de Project et Service.

   JSON

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticache:RemoveTagsFromResource",
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Service": "Elasticache",
                    "aws:ResourceTag/Project": "XYZ"
                },
                "ForAnyValue:StringNotEqualsIgnoreCase": {
                    "aws:TagKeys": [
                        "Project",
                        "Service"
                    ]
                }
            }
        }
    ]
}
```

3. Autoriser `AddTagsToResource` à n'importe quelle ressource uniquement si les balises sont différentes de Project et Service.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "elasticache:AddTagsToResource",
            "Resource": [
                "arn:aws:elasticache:*:*:*:*"
            ],
            "Condition": {
                "ForAnyValue:StringNotEqualsIgnoreCase": {
                    "aws:TagKeys": [
                        "Service",
                        "Project"
                    ]
```

```
                }
            }
        }
    ]
}
```

4. Refuser une action `CreateReplicationGroup` si la requête comporte `Tag Project=Foo`.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "elasticache:CreateReplicationGroup",
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": "Foo"
                }
            }
        }
    ]
}
```

5. Refuser une action `CopySnapshot` si l'instantané source contient la balise Project=XYZ et que la balise demandée est Service=ElastiCache.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "elasticache:CopySnapshot",
            "Resource": [
                "arn:aws:elasticache:*:*:snapshot:*"
            ],
            "Condition": {
                "StringEquals": {
```

```
                    "aws:ResourceTag/Project": "XYZ",
                    "aws:RequestTag/Service": "Elasticache"
                }
            }
        }
    ]
}
```

6. Refus de l'action `CreateCacheCluster` si l'identification de la demande `Project` est manquante ou n'est pas égale à `Dev`, `QA` ou `Prod`.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/Project": "true"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:AddTagsToResource"
            ],
            "Resource": "arn:aws:elasticache:*:*:cluster:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": [
                        "Dev",
                        "Prod",
                        "QA"
                    ]
                }
            }
        }
    ]
}
```

Pour obtenir des informations connexes sur les clés de condition, consultez Utilisation de clés de condition.

Surveillance des coûts avec des balises de répartition des coûts

Lorsque vous ajoutez des balises de répartition des coûts à vos ressources sur Amazon ElastiCache, vous pouvez suivre les coûts en regroupant les dépenses sur vos factures par valeur d'étiquette de ressource.

Une balise de répartition des ElastiCache coûts est une paire clé-valeur que vous définissez et associez à une ElastiCache ressource. 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 `ElastiCache` et `Asia Pacific (Singapore)`, et sur une autre ressource, les valeurs `ElastiCache` et `Europe (Frankfurt)`. Vous pouvez ensuite voir vos ElastiCache coûts totaux ventilés par région. Pour plus d'informations, consultez Utilisation des balises d'allocation des coûts dans le Guide de l'utilisateur AWS Billing .

Vous pouvez ajouter des balises de répartition des ElastiCache coûts aux clusters que vous avez ElastiCache conçus vous-même. Lorsque vous ajoutez, affichez, modifiez, copiez ou supprimez une balise, l'opération est appliquée uniquement au cluster spécifié.

Caractéristiques des étiquettes de répartition des ElastiCache coûts

- Les balises de répartition des coûts sont appliquées aux ElastiCache ressources spécifiées dans les opérations de la CLI et de l'API sous forme d'ARN. Le type de ressource sera un « cluster ».

  Exemple d'ARN : `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

  Exemple d'ARN : `arn:aws:elasticache:us-west-2:1234567890: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 ElastiCache ressource 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 `ElastiCache`.

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 ElastiCache ressource.

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.

Les balises de répartition des coûts sont appliquées aux ElastiCache clusters. Le cluster à étiqueter est spécifié à l'aide d'un ARN (Amazon Resource Name).

Exemple d'ARN : `arn:aws:elasticache:us-west-2:1234567890: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 pour lister une ElastiCache ressource existante en utilisant l'list-tags-for-resourceopération.

Le code suivant utilise le AWS CLI pour répertorier les balises du cluster Memcached `my-cluster` dans la région us-west-2.

Pour Linux, macOS ou Unix :

```
aws elasticache list-tags-for-resource \
```

```
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Pour Windows :

```
aws elasticache list-tags-for-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Le code suivant utilise le AWS CLI pour répertorier les balises du nœud Valkey ou Redis OSS du `my-cluster` cluster `my-cluster-001` dans la région us-west-2.

Pour Linux, macOS ou Unix :

```
aws elasticache list-tags-for-resource \
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Pour Windows :

```
aws elasticache list-tags-for-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

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 le AWS CLI pour ElastiCache list-tags-for-resource.

Ajout de balises à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour ajouter des balises à une ElastiCache ressource existante à l'aide de l'opération add-tags-to-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`elasticache`, `us-west-2` respectivement au nœud du cluster `my-cluster-001` `my-cluster` dans la région us-west-2.

Memcached

Pour Linux, macOS ou Unix :

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tags Key=Service,Value=elasticache \
        Key=Region,Value=us-west-2
```

Pour Windows :

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tags Key=Service,Value=elasticache ^
        Key=Region,Value=us-west-2
```

Redis

Pour Linux, macOS ou Unix :

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tags Key=Service,Value=elasticache \
        Key=Region,Value=us-west-2
```

Pour Windows :

```
aws elasticache add-tags-to-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tags Key=Service,Value=elasticache ^
        Key=Region,Value=us-west-2
```

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": "elasticache",
         "Key": "Service"
      },
      {
         "Value": "us-west-2",
         "Key": "Region"
      }
   ]
}
```

Pour plus d'informations, consultez le AWS CLI pour ElastiCache add-tags-to-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 cette opérationcreate-cache-cluster. Vous ne pouvez pas ajouter de balises lors de la création d'un cluster à l'aide de la console ElastiCache de gestion. Une fois que le cluster est créé, vous pouvez utiliser ensuite la console pour ajouter des balises au cluster.

Modification des balises à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour modifier les balises d'un ElastiCache cluster.

Pour modifier des balises :

- Utilisez add-tags-to-resource pour ajouter une nouvelle balise et une valeur, ou pour modifier la valeur associée à une balise existante.
- Utilisez remove-tags-from-resource pour supprimer des 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 les balises d'un cluster existant ElastiCache pour Memcached en utilisant l'remove-tags-from-resourceopération.

Pour Memcached, le code suivant utilise le AWS CLI pour supprimer les balises à l'aide des clés `Service` et `Region` du nœud du cluster `my-cluster-001` `my-cluster` dans la région us-west-2.

Pour Linux, macOS ou Unix :

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tag-keys PM Service
```

Pour Windows :

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tag-keys PM Service
```

Pour Redis OSS, le code suivant utilise le AWS CLI pour supprimer les balises à l'aide des clés `Service` et `Region` du nœud du cluster `my-cluster-001` `my-cluster` dans la région us-west-2.

Pour Linux, macOS ou Unix :

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tag-keys PM Service
```

Pour Windows :

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
 --tag-keys PM 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 pour ElastiCache remove-tags-from-resource.

Gestion de vos balises de répartition des coûts à l'aide de l' ElastiCache API

Vous pouvez utiliser l' ElastiCache API pour ajouter, modifier ou supprimer des balises de répartition des coûts.

Les balises de répartition des coûts sont appliquées ElastiCache aux clusters Memcached. Le cluster à étiqueter est spécifié à l'aide d'un ARN (Amazon Resource Name).

Exemple d'ARN : `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Rubriques

- Lister les tags à l'aide de l' ElastiCache API
- Ajouter des tags à l'aide de l' ElastiCache API
- Modification des balises à l'aide de l' ElastiCache API
- Supprimer des balises à l'aide de l' ElastiCache API

Lister les tags à l'aide de l' ElastiCache API

Vous pouvez utiliser l' ElastiCache API pour répertorier les balises d'une ressource existante à l'aide de l'ListTagsForResourceopération.

Pour Memcached, le code suivant utilise l' ElastiCache API pour répertorier les balises de la ressource `my-cluster` dans la région us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ListTagsForResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Pour Redis OSS, le code suivant utilise l' ElastiCache API pour répertorier les balises de la ressource `my-cluster-001` dans la région us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=ListTagsForResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Ajouter des tags à l'aide de l' ElastiCache API

Vous pouvez utiliser l' ElastiCache API pour ajouter des balises à un ElastiCache cluster existant à l'aide de l'AddTagsToResourceopération. 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' ElastiCache API pour ajouter les clés `Service` et `Region` avec les valeurs`elasticache`, `us-west-2` respectivement. Pour Memcached, cela s'applique à la ressource. `my-cluster` Pour Redis OSS, cela s'applique à la ressource de la `my-cluster-001` région us-west-2.

Memcached

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=AddTagsToResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=elasticache
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-west-2
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Redis

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=AddTagsToResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

```
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=elasticache
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-west-2
   &Version=2015-02-02
   &Timestamp=20150202T192317Z
   &X-Amz-Credential=<credential>
```

Pour plus d'informations, consultez AddTagsToResource le Amazon ElastiCache API Reference.

Modification des balises à l'aide de l' ElastiCache API

Vous pouvez utiliser l' ElastiCache API pour modifier les balises d'un ElastiCache cluster.

Pour modifier la valeur d'une balise :

- Utilisez l'opération AddTagsToResource pour ajouter une nouvelle balise et une valeur, ou pour modifier la valeur d'une balise existante.
- Utilisez RemoveTagsFromResource 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.

Utilisez RemoveTagsFromResource pour supprimer des balises de la ressource.

Supprimer des balises à l'aide de l' ElastiCache API

Vous pouvez utiliser l' ElastiCache API pour supprimer des balises d'un cluster existant ElastiCache pour Memcached en utilisant l'RemoveTagsFromResourceopération.

Le code suivant utilise l' ElastiCache API pour supprimer les balises à l'aide des clés `Service` et `Region` du nœud du cluster `my-cluster-001` `my-cluster` dans la région us-west-2.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=RemoveTagsFromResource
   &ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &TagKeys.member.1=Service
   &TagKeys.member.2=Region
```

```
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Utilisation de l'objectif Amazon ElastiCache Well-Architected

Cette section décrit l'Amazon ElastiCache Well-Architected Lens, un ensemble de principes de conception et de conseils pour concevoir des charges de travail bien architecturées. ElastiCache

- The ElastiCache Lens est un additif au AWS Well-Architected Framework.
- Chaque pilier comporte une série de questions pour aider à lancer la discussion autour d'une ElastiCache architecture.
  - Chaque question présente un certain nombre de meilleures pratiques ainsi que leurs scores pour la création d'un rapport.
    - Obligatoire : nécessaire avant de passer en production (si absent, risque élevé)
    - Meilleure pratique : le meilleur état possible pour un client
    - Bonne pratique : ce que nous recommandons aux clients (si absent, risque modéré)
- Terminologie Well-Architected
  - Composant : code, configuration et AWS ressources qui, ensemble, répondent à une exigence. Les composants interagissent avec d'autres composants et sont souvent assimilés à un service dans les architectures de microservices.
  - Charge de travail : ensemble de composants qui, ensemble, apportent de la valeur à l'entreprise. Des exemples de charges de travail sont les sites Web marketing, les sites Web d'e-commerce, les systèmes back-end pour une application mobile, les plateformes d'analyse, etc.

> **Note**
>
> Ce guide n'a pas été mis à jour pour inclure des informations sur la mise en cache ElastiCache sans serveur et le nouveau moteur Valkey.

Rubriques

- Pilier d'excellence opérationnelle d'Amazon ElastiCache Well-Architected Lens
- Pilier de ElastiCache sécurité Amazon Well-Architected Lens
- Pilier de fiabilité des objectifs Amazon ElastiCache Well-Architected

- Pilier d'efficacité des ElastiCache performances des objectifs Amazon Well-Architected
- Pilier d'optimisation des ElastiCache coûts d'objectif Amazon Well-Architected

Pilier d'excellence opérationnelle d'Amazon ElastiCache Well-Architected Lens

Le pilier d'excellence opérationnelle se concentre sur l'exécution et la surveillance des systèmes afin de générer de la valeur opérationnelle, ainsi que sur l'amélioration continue des processus et des procédures. Les sujets clés sont les suivants : automatisation des modifications, réponse aux événements et définition des normes pour gérer les opérations quotidiennes.

Rubriques

- OE 1 : Comment comprenez-vous les alertes et les événements déclenchés par votre ElastiCache cluster et comment y répondez-vous ?
- OE 2 : Quand et comment adaptez-vous vos ElastiCache clusters existants ?
- OE 3 : Comment gérez-vous les ressources de votre ElastiCache cluster et entretenez-vous votre cluster up-to-date ?
- OE 4 : Comment gérez-vous les connexions des clients à vos ElastiCache clusters ?
- OE 5 : Comment déployez-vous des ElastiCache composants pour une charge de travail ?
- EO 6 : Comment planifiez-vous et atténuez-vous les défaillances ?
- OE 7 : Comment résoudre les événements du moteur Valkey ou Redis OSS ?

OE 1 : Comment comprenez-vous les alertes et les événements déclenchés par votre ElastiCache cluster et comment y répondez-vous ?

Introduction au niveau des questions : Lorsque vous gérez des ElastiCache clusters, vous pouvez éventuellement recevoir des notifications et des alertes lorsque des événements spécifiques se produisent. ElastiCache, par défaut, enregistre les événements liés à vos ressources, tels qu'un basculement, le remplacement d'un nœud, une opération de dimensionnement, une maintenance planifiée, etc. Chaque événement inclut la date et l'heure, le nom et le type de source, ainsi qu'une description.

Avantage au niveau de la question : la capacité à comprendre et à gérer les raisons sous-jacentes des événements qui déclenchent les alertes générées par votre cluster vous permet de fonctionner plus efficacement et de répondre aux événements de manière appropriée.

- [Obligatoire] Passez en revue les événements générés ElastiCache sur la ElastiCache console (après avoir sélectionné votre région) ou à l'aide de la commande describe-events de l'interface de ligne de commande (AWS CLI) Amazon et de l'ElastiCache API. Configurez ElastiCache pour envoyer des notifications pour les événements importants du cluster à l'aide d'Amazon Simple Notification Service (Amazon SNS). L'utilisation d'Amazon SNS avec vos clusters vous permet de prendre des mesures programmatiques en cas d'événements. ElastiCache
  - Il existe deux grandes catégories d'événements : les événements en cours et les événements planifiés. La liste des événements actuels inclut : la création et la suppression de ressources, les opérations de mise à l'échelle, le basculement, le redémarrage du nœud, la création d'un instantané, la modification d'un paramètre du cluster, le renouvellement d'un certificat CA, les événements d'échec (échec de l'allocation du cluster (VPC ou ENI), échec de mise à l'échelle (ENI) et échec d'un instantané). La liste des événements planifiés inclut : nœud dont le remplacement est planifié pendant la fenêtre de maintenance et nœud dont le remplacement a été replanifié.
  - Bien que vous n'ayez pas besoin de réagir immédiatement à certains de ces événements, il est essentiel de commencer par examiner tous les événements d'échec :
    - ElastiCache:AddCacheNodeFailed
    - ElastiCache:CacheClusterProvisioningFailed
    - ElastiCache:CacheClusterScalingFailed
    - ElastiCache:CacheNodesRebooted
    - ElastiCache: SnapshotFailed (Valkey ou Redis OSS uniquement)
  - [Ressources] :
    - Gestion des ElastiCache notifications Amazon SNS
    - Notifications d'événements Amazon SNS
- [Idéal] Pour automatiser les réponses aux événements, tirez parti des fonctionnalités AWS des produits et services telles que les fonctions SNS et Lambda. Suivez les meilleures pratiques en apportant des modifications mineures, fréquentes et réversibles, sous forme de code pour faire évoluer vos opérations au fil du temps. Vous devez utiliser CloudWatch les métriques Amazon pour surveiller vos clusters.

  [Ressources] : Surveillez ElastiCache (mode cluster désactivé) lisez les points de terminaison des répliques à l'aide de AWS Lambda, Amazon Route 53 et Amazon SNS pour un cas d'utilisation utilisant Lambda et SNS.

OE 2 : Quand et comment adaptez-vous vos ElastiCache clusters existants ?

Introduction au niveau des questions : Le dimensionnement correct de votre ElastiCache cluster est un exercice d'équilibre qui doit être évalué chaque fois que des modifications sont apportées aux types de charge de travail sous-jacents. Votre objectif est de fonctionner avec un environnement de taille adaptée à votre charge de travail.

Avantage au niveau de la question : une utilisation excessive de vos ressources peut entraîner une latence élevée et une diminution globale des performances. La sous-utilisation, en revanche, peut entraîner une sur-allocation des ressources avec une optimisation des coûts non optimale. En ajustant la taille de vos environnements, vous pouvez trouver un équilibre entre efficacité des performances et optimisation des coûts. Pour remédier à la surutilisation ou à la sous-utilisation de vos ressources, vous ElastiCache pouvez effectuer une mise à l'échelle en deux dimensions. Vous pouvez effectuer une mise à l'échelle verticale en augmentant ou en diminuant la capacité des nœuds. Vous pouvez également effectuer une mise à l'échelle horizontale en ajoutant et en supprimant des nœuds.

- [Obligatoire] La surutilisation de l'UC et du réseau sur les nœuds primaires doit être corrigée en déchargeant et en redirigeant les opérations de lecture vers les nœuds de réplicas. Utilisez des nœuds de réplicas pour les opérations de lecture afin de réduire l'utilisation du nœud primaire. Cela peut être configuré dans votre bibliothèque cliente Valkey ou Redis OSS en vous connectant au point de terminaison du ElastiCache lecteur pour désactiver le mode cluster, ou en utilisant la commande READONLY pour le mode cluster activé.

  [Ressources] :

  - Recherche de points de terminaison de connexion dans ElastiCache
  - Cluster Right-Sizing
  - Commande READONLY
- [Obligatoire] Surveillez l'utilisation des ressources critiques du cluster telles que l'UC, la mémoire et le réseau. L'utilisation de ces ressources de cluster spécifiques doit faire l'objet d'un suivi afin de prendre des décisions éclairées quant à la mise à l'échelle et le type d'opération de mise à l'échelle. Lorsque le mode ElastiCache cluster est désactivé, les nœuds principaux et répliques peuvent être redimensionnés verticalement. Les nœuds de réplicas peuvent également être mis à l'échelle horizontalement de 0 à 5 nœuds. Si le mode cluster est activé, il en va de même pour chaque partition de votre cluster. En outre, vous pouvez augmenter ou réduire le nombre de partitions.

[Ressources] :

- Surveillance des meilleures pratiques liées à ElastiCache l'utilisation d'Amazon CloudWatch
- Mise à l'échelle des ElastiCache clusters pour Valkey et Redis OSS
- Mise à l'échelle ElastiCache des clusters pour Memcached

- [Meilleure pratique] La surveillance des tendances au fil du temps peut vous aider à détecter les modifications de la charge de travail qui passeraient inaperçues si elles étaient surveillées à un moment donné. Pour détecter les tendances à long terme, utilisez CloudWatch des métriques pour rechercher des plages temporelles plus longues. Les enseignements tirés de l'observation de longues périodes de CloudWatch mesures devraient éclairer vos prévisions concernant l'utilisation des ressources du cluster. CloudWatch les points de données et les mesures sont disponibles pendant 455 jours au maximum.

  [Ressources] :

  - Surveillance à ElastiCache l'aide de CloudWatch métriques
  - Surveillance de Memcached avec des métriques CloudWatch
  - Surveillance des meilleures pratiques liées à ElastiCache l'utilisation d'Amazon CloudWatch

- [Idéal] Si vos ElastiCache ressources sont créées avec, CloudFormation il est recommandé d'effectuer des modifications à l'aide de CloudFormation modèles afin de préserver la cohérence opérationnelle et d'éviter les modifications de configuration non gérées et les dérives de pile.

  [Ressources] :

  - ElastiCache référence de type de ressource pour CloudFormation

- [Mieux] Automatisez vos opérations de dimensionnement à l'aide des données opérationnelles du cluster et définissez des seuils CloudWatch pour configurer les alarmes. Utilisez CloudWatch Events and Simple Notification Service (SNS) pour déclencher des fonctions Lambda et exécuter une ElastiCache API afin de dimensionner automatiquement vos clusters. Par exemple, vous pouvez ajouter une partition à votre cluster lorsque la métrique `EngineCPUUtilization` atteint 80 % pendant une période prolongée. Une autre option serait d'utiliser `DatabaseMemoryUsedPercentages` pour un seuil basé sur la mémoire.

  [Ressources] :

  - Utilisation d'Amazon CloudWatch Alarms
  - Que sont les CloudWatch événements Amazon ?
  - Utilisation AWS Lambda avec Amazon Simple Notification Service

- Référence d'API ElastiCache

OE 3 : Comment gérez-vous les ressources de votre ElastiCache cluster et entretenez-vous votre cluster up-to-date ?

Introduction au niveau des questions : Lorsque vous opérez à grande échelle, il est essentiel que vous soyez en mesure d'identifier et d'identifier toutes vos ressources. ElastiCache Lorsque vous déployez de nouvelles fonctionnalités d'application, vous devez créer une symétrie entre les versions de cluster dans tous vos types d' ElastiCache environnement : développement, test et production. Les attributs de ressources vous permettent de séparer les environnements pour différents objectifs opérationnels, par exemple lors du déploiement de nouvelles fonctionnalités et de l'activation de nouveaux mécanismes de sécurité.

Avantage au niveau de la question : la meilleure pratique opérationnelle consiste à séparer vos environnements de développement, de test et de production. Il est également recommandé que les derniers correctifs logiciels soient appliqués à vos clusters et nœuds dans tous les environnements à l'aide de processus bien compris et documentés. Tirer parti des ElastiCache fonctionnalités natives permet à votre équipe d'ingénierie de se concentrer sur l'atteinte des objectifs commerciaux et non sur la ElastiCache maintenance.

- [Mieux] Utilisez la dernière version du moteur disponible et appliquez les mises à jour en libre-service dès qu'elles sont disponibles. ElastiCache met automatiquement à jour son infrastructure sous-jacente pendant la période de maintenance que vous avez spécifiée pour le cluster. Toutefois, les nœuds qui s'exécutent dans vos clusters sont mis à jour via les mises à jour en libre-service. Ces mises à jour peuvent être de deux types : correctifs de sécurité ou mises à jour logicielles mineures. Assurez-vous de bien comprendre la différence entre les types de correctifs et le moment où ils sont appliqués.

  [Ressources] :

  - Mises à jour en libre-service sur Amazon ElastiCache
  - Page d'aide ElastiCache relative à la maintenance gérée et aux mises à jour des services Amazon
- [Idéal] Organisez vos ElastiCache ressources à l'aide de balises. Utilisez des balises sur des groupes de réplication et non sur des nœuds individuels. Vous pouvez configurer les balises à afficher lorsque vous interrogez des ressources et vous pouvez utiliser des balises pour effectuer des recherches et appliquer des filtres. Vous devez utiliser les groupes de ressources pour créer et gérer facilement des collections de ressources qui partagent des ensembles de balises communs.

[Ressources] :

- Tagging Best Practices
- ElastiCache référence de type de ressource pour CloudFormation
- Groupes de paramètres

OE 4 : Comment gérez-vous les connexions des clients à vos ElastiCache clusters ?

Introduction au niveau des questions : lorsque vous opérez à grande échelle, vous devez comprendre comment vos clients se connectent au ElastiCache cluster pour gérer les aspects opérationnels de votre application (tels que les temps de réponse).

Avantage au niveau de la question : le choix du mécanisme de connexion le plus approprié garantit que votre application ne se déconnecte pas en raison d'erreurs de connectivité, telles que des délais d'expiration.

- [Obligatoire] Séparez les opérations de lecture des opérations d'écriture et connectez-vous aux nœuds de réplicas pour exécuter les opérations de lecture. Sachez toutefois que lorsque vous séparez les écritures des lectures, vous perdrez la possibilité de lire une clé immédiatement après l'avoir écrite en raison de la nature asynchrone de la réplication Valkey et Redis OSS. La commande WAIT peut être utilisée pour améliorer la sécurité des données réelles et forcer les réplicas à accuser réception des écritures avant de répondre aux clients, moyennant un coût global en termes de performances. L'utilisation de nœuds de réplication pour les opérations de lecture peut être configurée dans votre bibliothèque ElastiCache cliente à l'aide du point de terminaison du ElastiCache lecteur pour désactiver le mode cluster. Pour activer le mode cluster, utilisez la commande READONLY. Pour de nombreuses bibliothèques ElastiCache clientes, READONLY est implémenté par défaut ou via un paramètre de configuration.

  [Ressources] :

  - Recherche de points de terminaison de connexion dans ElastiCache
  - READONLY
- [Obligatoire] Utilisez le regroupement de connexions. L'établissement d'une connexion TCP entraîne des coûts en termes de temps UC, à la fois côté client et côté serveur, et le regroupement permet de réutiliser la connexion TCP.

  Pour réduire la surcharge de connexion, vous devez utiliser le regroupement de connexions. Avec un pool de connexions, votre application peut réutiliser et libérer des connexions « à volonté »,

sans les coûts liés à l'établissement de la connexion. Vous pouvez implémenter le regroupement de connexions via votre bibliothèque ElastiCache cliente (si elle est prise en charge), avec un framework disponible pour votre environnement d'application, ou le créer à partir de zéro.

- [Meilleure pratique] Assurez-vous que le délai d'expiration du socket du client est défini sur au moins une seconde (la valeur par défaut « aucun » est généralement définie dans certains clients).
  - Si la valeur du délai d'attente est trop faible, cela peut entraîner des délais d'expiration lorsque la charge du serveur est élevée. Si cette valeur est trop élevée, votre application peut mettre du temps à détecter les problèmes de connexion.
  - Contrôlez le volume des nouvelles connexions en implémentant le regroupement de connexions dans votre application client. Cela réduit la latence et l'utilisation de l'UC nécessaires à l'ouverture et à la fermeture de connexions, et à l'établissement d'une liaison TLS si le protocole TLS est activé sur le cluster.

  [Ressources] : Configuration ElastiCache pour une meilleure disponibilité
- [Bonne pratique] La mise en pipeline (lorsque vos cas d'utilisation le permettent) peut améliorer les performances de manière significative.
  - Grâce à la mise en pipeline, vous réduisez le temps d'aller-retour (RTT) entre vos clients d'applications et le cluster et les nouvelles demandes peuvent être traitées même si le client n'a pas encore lu les réponses précédentes.
  - Avec la mise en pipeline, vous pouvez envoyer plusieurs commandes au serveur sans attendre de réponse/accusé de réception. La mise en pipeline a toutefois un inconvénient : lorsque vous finissez par extraire toutes les réponses en bloc, il se peut qu'une erreur se soit produite et que vous ne la détectiez pas avant la fin.
  - Implémentez des méthodes pour renouveler les demandes lorsque l'erreur renvoyée omet la mauvaise demande.

  [Ressources] : Pipelining

OE 5 : Comment déployez-vous des ElastiCache composants pour une charge de travail ?

Introduction au niveau des questions : ElastiCache les environnements peuvent être déployés manuellement via la AWS console, ou par programmation via une APIs CLI, des boîtes à outils, etc. Les meilleures pratiques en matière d'excellence opérationnelle suggèrent d'automatiser les déploiements par le biais du code chaque fois que cela est possible. En outre, les ElastiCache clusters peuvent être isolés par charge de travail ou combinés à des fins d'optimisation des coûts.

Avantage au niveau des questions : le choix du mécanisme de déploiement le mieux adapté à vos ElastiCache environnements peut améliorer l'excellence opérationnelle au fil du temps. Il est recommandé d'effectuer les opérations sous forme de code dans la mesure du possible afin de minimiser les erreurs humaines et d'augmenter la reproductibilité, la flexibilité et le temps de réponse aux événements.

En comprenant les exigences d'isolation des charges de travail, vous pouvez choisir d'avoir ElastiCache des environnements dédiés par charge de travail ou de combiner plusieurs charges de travail en clusters uniques, ou des combinaisons des deux. Comprendre les compromis peut aider à trouver un équilibre entre excellence opérationnelle et optimisation des coûts.

- [Obligatoire] Comprenez les options de déploiement disponibles et automatisez ces procédures dans la mesure du possible. ElastiCache Les voies d'automatisation possibles incluent CloudFormation AWS CLI/SDK et. APIs

  [Ressources] :

  - Référence du type de ElastiCache ressource Amazon
  - elasticache
  - Référence ElastiCache d'API Amazon
- [Obligatoire] Pour toutes les charges de travail, déterminez le niveau d'isolation du cluster requis.
  - [Meilleure pratique] : Isolation élevée : mappage 1:1 de la charge de travail au cluster. Permet un contrôle précis de l'accès, du dimensionnement, de la mise à l'échelle et de la gestion des ElastiCache ressources par charge de travail.
  - [Pratique encore meilleure] : Isolation moyenne : M:1 isolé par objectif, mais peut-être partagé entre plusieurs charges de travail (par exemple, un cluster dédié à la mise en cache des charges de travail et un autre dédié à la messagerie).
  - [Bonne pratique] : Faible isolation : M:1 polyvalent, entièrement partagé. Recommandé pour les charges de travail où l'accès partagé est admis.

EO 6 : Comment planifiez-vous et atténuez-vous les défaillances ?

Introduction au niveau des questions : L'excellence opérationnelle consiste notamment à anticiper les défaillances en effectuant régulièrement des exercices « pré-mortem » afin d'identifier les sources potentielles de défaillance afin de les éliminer ou de les atténuer. ElastiCache propose une API Failover qui permet de simuler des événements de défaillance d'un nœud, à des fins de test.

Avantage au niveau de la question : en testant les scénarios de défaillance à l'avance, vous pouvez découvrir leur impact sur votre charge de travail. Cela permet de tester en toute sécurité les procédures de réponse et leur efficacité, tout en familiarisant votre équipe avec leur exécution.

[Obligatoire] Effectuez régulièrement des tests de basculement dans les comptes de développement/test. TestFailover

OE 7 : Comment résoudre les événements du moteur Valkey ou Redis OSS ?

Introduction au niveau des questions : l'excellence opérationnelle nécessite la capacité d'étudier à la fois les informations relatives au niveau du service et au niveau du moteur afin d'analyser l'état et l'état de vos clusters. ElastiCache peut émettre des journaux du moteur Valkey ou Redis OSS à la fois vers Amazon CloudWatch et Amazon Kinesis Data Firehose.

Avantage au niveau des questions : l'activation des journaux du moteur Valkey ou Redis OSS sur les ElastiCache clusters permet de mieux comprendre les événements qui ont un impact sur l'état et les performances des clusters. Les journaux du moteur Valkey ou Redis OSS fournissent des données directement à partir du moteur qui ne sont pas disponibles via le mécanisme ElastiCache des événements. En observant attentivement les ElastiCache événements (voir OE-1 précédent) et les journaux du moteur, il est possible de déterminer l'ordre des événements lors du dépannage du point de vue du ElastiCache service et du point de vue du moteur.

- [Obligatoire] Assurez-vous que la fonctionnalité de journalisation du moteur Redis OSS est activée, qui est disponible depuis la ElastiCache version 6.2 pour Redis OSS et les versions ultérieures. Vous pouvez le faire lors de la création du cluster ou en modifiant le cluster après sa création.
  - Déterminez si Amazon CloudWatch Logs ou Amazon Kinesis Data Firehose est la cible appropriée pour les journaux du moteur Redis OSS.
  - Sélectionnez un journal cible approprié dans Kinesis Data Firehose CloudWatch ou dans Kinesis Data Firehose pour conserver les journaux. Si vous possédez plusieurs clusters, envisagez d'utiliser un journal cible différent pour chaque cluster. Cela vous aidera à isoler les données lors de la résolution des problèmes.

  [Ressources] :

  - Livraison de journaux
  - Destinations de journalisation : Amazon CloudWatch Logs
  - Présentation d'Amazon CloudWatch Logs : Qu'est-ce qu'Amazon CloudWatch Logs ?
  - Présentation d'Amazon Kinesis Data Firehose : What Is Amazon Kinesis Data Firehose?

- [Mieux] Si vous utilisez Amazon CloudWatch Logs, pensez à utiliser Amazon CloudWatch Logs Insights pour interroger le journal du moteur Valkey ou Redis OSS afin d'obtenir des informations importantes.

  Par exemple, créez une requête CloudWatch sur le groupe de journaux contenant les journaux du moteur Valkey ou Redis OSS qui renverra des événements marqués LogLevel d'un « AVERTISSEMENT », tels que :

  ```
  fields @timestamp, LogLevel, Message
  | sort @timestamp desc
  | filter LogLevel = "WARNING"
  ```

  [Ressources] : Analyse des données des CloudWatch journaux avec Logs Insights

Pilier de ElastiCache sécurité Amazon Well-Architected Lens

Le pilier de sécurité se concentre sur la protection des informations et des systèmes. Les sujets clés sont les suivants : confidentialité et intégrité des données, identification et gestion de qui peut faire quoi grâce à la gestion basée sur les privilèges, protection des systèmes et mise en place de contrôles pour détecter les événements de sécurité.

Rubriques

- SEC 1 : Quelles mesures prenez-vous pour contrôler l'accès autorisé aux ElastiCache données ?
- SEC 2 : Vos applications nécessitent-elles une autorisation supplémentaire au-delà ElastiCache des contrôles basés sur le réseau ?
- PS 3 : Existe-t-il un risque que des commandes soient exécutées par inadvertance, entraînant une perte ou une défaillance des données ?
- SEC 4 : Comment garantir le chiffrement des données au repos avec ElastiCache
- SEC 5 : Comment cryptez-vous les données en transit ? ElastiCache
- PS 6 : Comment restreignez-vous l'accès aux ressources du plan de contrôle ?
- PS 7 : Comment détectez-vous les événements de sécurité et comment y répondez-vous ?

SEC 1 : Quelles mesures prenez-vous pour contrôler l'accès autorisé aux ElastiCache données ?

Introduction au niveau des questions : tous les ElastiCache clusters sont conçus pour être accessibles depuis des instances Amazon Elastic Compute Cloud dans un VPC, des fonctions sans serveur (AWS Lambda) ou des conteneurs (Amazon Elastic Container Service). Le scénario le plus courant consiste à accéder à un ElastiCache cluster depuis une instance Amazon Elastic Compute Cloud au sein du même Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Avant de pouvoir vous connecter à un cluster depuis une EC2 instance Amazon, vous devez autoriser l' EC2 instance Amazon à accéder au cluster. Pour accéder à un ElastiCache cluster exécuté dans un VPC, il est nécessaire d'autoriser l'accès réseau au cluster.

Avantage au niveau de la question : l'entrée réseau dans le cluster est contrôlée via les groupes de sécurité du VPC. Un groupe de sécurité agit comme un pare-feu virtuel pour vos EC2 instances Amazon afin de contrôler le trafic entrant et sortant. Les règles entrantes contrôlent le trafic entrant vers votre instance, et les règles sortantes contrôlent le trafic sortant de votre instance. Dans le cas où ElastiCache, lors du lancement d'un cluster, il est nécessaire d'associer un groupe de sécurité. Cela garantit que les règles de trafic entrant et sortant sont en place pour tous les nœuds qui composent le cluster. En outre, ElastiCache il est configuré pour être déployé exclusivement sur des sous-réseaux privés, de sorte qu'ils ne soient accessibles que via le réseau privé du VPC.

- [Obligatoire] Le groupe de sécurité associé à votre cluster contrôle l'entrée réseau et l'accès au cluster. Par défaut, aucune règle entrante n'est définie pour un groupe de sécurité et, par conséquent, aucun chemin d'entrée vers celui-ci ne sera défini. ElastiCache Pour activer cela, configurez une règle entrante sur le groupe de sécurité spécifiant l'adresse/plage IP source, le trafic de type TCP et le port de votre ElastiCache cluster (port par défaut 6379 pour Valkey et Redis ElastiCache OSS par exemple). Bien qu'il soit possible d'autoriser un très large éventail de sources d'entrée, comme toutes les ressources d'un VPC (0.0.0.0/0), il est conseillé d'être aussi précis que possible dans la définition des règles entrantes, par exemple en autorisant uniquement l'accès entrant aux clients Valkey ou Redis OSS exécutés sur des instances Amazon Amazon associées à un groupe de sécurité spécifique. EC2

  [Ressources] :

  - Sous-réseaux et groupes de sous-réseaux
  - Accès à votre cluster ou groupe de réplication
  - Contrôler le trafic vers vos ressources AWS à l'aide de groupes de sécurité
  - Groupes de sécurité Amazon EC2 pour les instances Linux

- Des AWS Identity and Access Management politiques [obligatoires] peuvent être attribuées à des AWS Lambda fonctions leur permettant d'accéder aux ElastiCache données. Pour activer cette fonctionnalité, créez un rôle d'exécution IAM avec l'`AWSLambdaVPCAccessExecutionRole`autorisation, puis attribuez le rôle à la AWS Lambda fonction.

  [Ressources] : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un Amazon VPC : Tutoriel : Configuration d'une fonction Lambda pour accéder à Amazon dans un Amazon VPC ElastiCache

SEC 2 : Vos applications nécessitent-elles une autorisation supplémentaire au-delà ElastiCache des contrôles basés sur le réseau ?

Introduction au niveau des questions : dans les scénarios où il est nécessaire de restreindre ou de contrôler l'accès aux clusters au niveau d'un client individuel, il est recommandé de s'authentifier via la commande AUTH. ElastiCache les jetons d'authentification, avec gestion optionnelle des utilisateurs et des groupes d'utilisateurs, permettent ElastiCache d'exiger un mot de passe avant de permettre aux clients d'exécuter des commandes et des clés d'accès, améliorant ainsi la sécurité du plan de données.

Avantage au niveau des questions : pour garantir la sécurité de vos données, ElastiCache fournit des mécanismes de protection contre tout accès non autorisé à vos données. Cela inclut l'application du contrôle d'accès basé sur les rôles (RBAC) AUTH ou du jeton AUTH (mot de passe) auquel les clients doivent se connecter avant d'exécuter des commandes autorisées. ElastiCache

- [Idéal] Pour les ElastiCache versions 6.x et supérieures pour Redis OSS, et les ElastiCache versions 7.2 et supérieures pour Valkey, définissez les contrôles d'authentification et d'autorisation en définissant les groupes d'utilisateurs, les utilisateurs et les chaînes d'accès. Attribuez des utilisateurs à des groupes d'utilisateurs, puis attribuez des groupes d'utilisateurs à des clusters. Pour utiliser le RBAC, il doit être sélectionné lors de la création du cluster et le chiffrement en transit doit être activé. Assurez-vous d'utiliser un client Valkey ou Redis OSS compatible avec le protocole TLS pour pouvoir tirer parti du RBAC.

  [Ressources] :

  - Appliquer le RBAC à un groupe de réplication pour ElastiCache
  - Définition des autorisations à l'aide d'une chaîne d'accès
  - ACL

- ElastiCache Versions prises en charge
- [Idéal] Pour ElastiCache les versions antérieures à la version 6.x pour Redis OSS, en plus de définir une politique de mot de passe stricte token/password et de maintenir une politique de mot de passe stricte pour AUTH, il est recommandé de faire pivoter le mot de passe/jeton. ElastiCache peut gérer jusqu'à deux (2) jetons d'authentification à la fois. Vous pouvez également modifier le cluster pour exiger explicitement l'utilisation de jetons d'authentification.

  [Ressources] : Modification du jeton AUTH sur un cluster existant ElastiCache

PS 3 : Existe-t-il un risque que des commandes soient exécutées par inadvertance, entraînant une perte ou une défaillance des données ?

Introduction au niveau des questions : Un certain nombre de commandes Valkey ou Redis OSS peuvent avoir un impact négatif sur les opérations si elles sont exécutées par erreur ou par des acteurs malveillants. Ces commandes peuvent avoir des conséquences imprévues du point de vue des performances et de la sécurité des données. Par exemple, un développeur peut appeler régulièrement la commande FLUSHALL dans un environnement de développement et, en raison d'une erreur, peut tenter par inadvertance d'appeler cette commande sur un système de production, ce qui entraîne une perte de données accidentelle.

Avantage au niveau des questions : à partir de ElastiCache la version 5.0.3 pour Redis OSS, il est possible de renommer certaines commandes susceptibles de perturber votre charge de travail. Le fait de renommer les commandes permet d'éviter qu'elles ne soient exécutées par inadvertance sur le cluster.

- [Obligatoire]

  [Ressources] :

  - ElastiCache version 5.0.3 pour Redis OSS (obsolète, utilisez la version 5.0.6)
  - ElastiCache version 5.0.3 pour les modifications des paramètres de Redis OSS
  - Sécurité Redis OSS

SEC 4 : Comment garantir le chiffrement des données au repos avec ElastiCache

Introduction au niveau des questions : Bien qu'il ElastiCache s'agisse d'un magasin de données en mémoire, il est possible de chiffrer toutes les données susceptibles d'être conservées (sur le stockage) dans le cadre des opérations standard du cluster. Cela inclut à la fois les sauvegardes

planifiées et manuelles écrites sur Amazon S3, mais également les données enregistrées dans le stockage sur disque à la suite d'opérations de synchronisation et d'échange. Les types d'instances des familles M6g et R6g proposent également le chiffrement en mémoire permanent.

Avantage au niveau des questions : ElastiCache fournit un chiffrement au repos en option pour renforcer la sécurité des données.

- [Obligatoire] Le chiffrement au repos ne peut être activé sur un ElastiCache cluster (groupe de réplication) que lors de sa création. Un cluster existant ne peut pas être modifié pour commencer à chiffrer les données au repos. Par défaut, ElastiCache fournira et gérera les clés utilisées pour le chiffrement au repos.

  [Ressources] :

  - Contraintes de chiffrement au repos
  - Activation du chiffrement au repos
- [Mieux] Tirez parti des types d' EC2 instances Amazon qui chiffrent les données lorsqu'elles sont en mémoire (comme M6g ou R6g). Dans la mesure du possible, envisagez de gérer vos propres clés pour le chiffrement au repos. Pour les environnements de sécurité des données plus stricts, AWS Key Management Service (KMS) peut être utilisé pour gérer automatiquement les clés principales du client (CMK). Grâce à ElastiCache l'intégration avec AWS Key Management Service, vous pouvez créer, posséder et gérer les clés utilisées pour le chiffrement des données au repos pour votre ElastiCache cluster.

  [Ressources] :

  - Utilisation de clés gérées par le client depuis AWS Key Management Service
  - AWS Service de gestion des clés
  - AWS KMS concepts

SEC 5 : Comment cryptez-vous les données en transit ? ElastiCache

Introduction au niveau de la question : il est souvent exigé d'éviter que les données ne soient compromises pendant leur transit. Cela représente les données au sein des composants d'un système distribué, ainsi qu'entre les clients d'applications et les nœuds de cluster. ElastiCache répond à cette exigence en permettant le chiffrement des données en transit entre les clients et le cluster, et entre les nœuds du cluster eux-mêmes. Les types d'instances des familles M6g et R6g proposent également le chiffrement en mémoire permanent.

Avantage détaillé : le chiffrement ElastiCache en transit d'Amazon est une fonctionnalité facultative qui vous permet de renforcer la sécurité de vos données aux points les plus vulnérables, lorsqu'elles sont en transit d'un endroit à un autre.

- [Obligatoire] Le chiffrement en transit ne peut être activé que sur un cluster (groupe de réplication) lors de sa création. Veuillez noter qu'en raison du traitement supplémentaire requis pour le chiffrement/déchiffrement des données, la mise en œuvre du chiffrement en transit aura un certain impact sur les performances. Pour comprendre l'impact, il est recommandé de comparer votre charge de travail avant et après l'activationencryption-in-transit.

  [Ressources] :

  - Présentation du chiffrement en transit

PS 6 : Comment restreignez-vous l'accès aux ressources du plan de contrôle ?

Introduction au niveau des questions : les politiques IAM et l'ARN permettent des contrôles d'accès précis ElastiCache pour Valkey et Redis OSS, permettant ainsi un contrôle plus strict de la gestion de la création, de la modification et de la suppression des clusters.

Avantage au niveau des questions : la gestion des ElastiCache ressources Amazon, telles que les groupes de réplication, les nœuds, etc., peut être limitée aux AWS comptes dotés d'autorisations spécifiques basées sur les politiques IAM, ce qui améliore la sécurité et la fiabilité des ressources.

- [Obligatoire] Gérez l'accès aux ElastiCache ressources Amazon en attribuant des AWS Identity and Access Management politiques spécifiques aux AWS utilisateurs, afin de mieux contrôler quels comptes peuvent effectuer quelles actions sur les clusters.

  [Ressources] :

  - Vue d'ensemble de la gestion des autorisations d'accès à vos ElastiCache ressources
  - Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon ElastiCache

PS 7 : Comment détectez-vous les événements de sécurité et comment y répondez-vous ?

Introduction au niveau des questions : ElastiCache lorsqu'il est déployé avec le RBAC activé, exporte des CloudWatch métriques pour informer les utilisateurs des événements de sécurité. Ces métriques

permettent d'identifier les tentatives infructueuses d'authentification, d'accès aux clés ou d'exécution des commandes que les utilisateurs RBAC connectés ne sont pas autorisés à effectuer.

En outre, AWS les ressources relatives aux produits et services contribuent à sécuriser votre charge de travail globale en automatisant les déploiements et en enregistrant toutes les actions et modifications pour un examen ou un audit ultérieurs.

Avantage au niveau de la question : en surveillant les événements, vous permettez à votre organisation de répondre conformément à vos exigences, vos politiques et vos procédures. L'automatisation de la surveillance et des réponses à ces événements de sécurité renforce votre posture de sécurité globale.

- [Obligatoire] Familiarisez-vous avec les CloudWatch métriques publiées concernant les échecs d'authentification et d'autorisation RBAC.
  - AuthenticationFailures = Tentatives d'authentification infructueuses auprès de Valkey ou Redis OSS
  - KeyAuthorizationFailures = Tentatives infructueuses des utilisateurs pour accéder aux clés sans autorisation
  - CommandAuthorizationFailures = Tentatives infructueuses d'exécution de commandes par des utilisateurs sans autorisation

  [Ressources] :

  - Métriques pour Valkey ou Redis OSS
- [Meilleure pratique] Il est recommandé de configurer des alertes et des notifications sur ces métriques et d'y répondre si nécessaire.

  [Ressources] :

  - Utilisation des CloudWatch alarmes Amazon
- [Mieux] Utilisez la commande Valkey ou Redis OSS ACL LOG pour recueillir plus de détails

  [Ressources] :

  - ACL LOG
- [Mieux] Familiarisez-vous avec les fonctionnalités AWS des produits et services en matière de surveillance, de journalisation et d'analyse des ElastiCache déploiements et des événements

  [Ressources] :

  - Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail

- elasticache-redis-cluster-automatic-vérification des sauvegardes
- Surveillance de l'utilisation à l'aide de CloudWatch métriques

Pilier de fiabilité des objectifs Amazon ElastiCache Well-Architected

Le pilier de la fiabilité met l'accent sur les charges de travail exécutant les fonctions prévues et sur la manière de se remettre rapidement en cas d'incapacité à répondre aux demandes. Les sujets clés incluent la conception de systèmes distribués, la planification de la restauration et l'adaptation à l'évolution des exigences.

Rubriques

- PF 1 : Comment prenez-vous en charge les déploiements d'architecture haute disponibilité ?
- REL 2 : Comment atteignez-vous vos objectifs de point de reprise (RPOs) ElastiCache ?
- PF 3 : Comment répondez-vous aux exigences de reprise après sinistre (DR) ?
- PF 4 : Comment planifiez-vous efficacement les basculements ?
- REL 5 : Vos ElastiCache composants sont-ils conçus pour évoluer ?

PF 1 : Comment prenez-vous en charge les déploiements d'architecture haute disponibilité ?

Introduction au niveau des questions : Comprendre l'architecture de haute disponibilité d'Amazon vous ElastiCache permettra de fonctionner de manière résiliente lors d'événements de disponibilité.

Avantage au niveau des questions : l'architecture de vos ElastiCache clusters de manière à ce qu'ils soient résilients aux défaillances garantit une meilleure disponibilité de vos déploiements. ElastiCache

- [Obligatoire] Déterminez le niveau de fiabilité dont vous avez besoin pour votre ElastiCache cluster. Les différentes charges de travail sont soumises à des normes de résilience différentes, qu'il s'agisse de charges de travail entièrement éphémères ou de charges de travail essentielles à la mission. Définissez les besoins pour chaque type d'environnement que vous exploitez, tel que le développement, le test et la production.

  Moteur de mise en cache : ElastiCache pour Memcached ou ElastiCache pour Valkey et Redis OSS

1. ElastiCache for Memcached ne fournit aucun mécanisme de réplication et est principalement utilisé pour les charges de travail éphémères.
2. ElastiCache pour Valkey et Redis, OSS propose des fonctionnalités HA décrites ci-dessous

- [Idéal] Pour les charges de travail nécessitant une haute disponibilité, utilisez-les ElastiCache en mode cluster avec un minimum de deux répliques par partition, même pour les charges de travail nécessitant un faible débit qui ne nécessitent qu'une seule partition.

1. Lorsque le mode cluster est activé, la configuration Multi-AZ est activée automatiquement.

   Multi-AZ minimise les interruptions en effectuant des basculements automatiques du nœud primaire vers les réplicas, en cas de maintenance planifiée ou non planifiée, et en atténuant les défaillances de la zone de disponibilité.
2. Pour les charges de travail partitionnées, un minimum de trois partitions permet une restauration plus rapide en cas de basculement, car le protocole de cluster Valkey ou Redis OSS exige que la majorité des nœuds principaux soient disponibles pour atteindre le quorum.
3. Configurez deux réplicas ou plus selon la disponibilité.

   Le fait de disposer de deux réplicas améliore la capacité de mise à l'échelle en lecture ainsi que la disponibilité en lecture dans les scénarios où un réplica est en cours de maintenance.
4. Utilisez des types de nœuds basés sur Graviton2 (nœuds par défaut dans la plupart des régions).

   ElastiCache a ajouté des performances optimisées sur ces nœuds. Vous bénéficiez ainsi de meilleures performances en termes de réplication et de synchronisation, ce qui se traduit par une disponibilité globale améliorée.
5. Surveillez et ajustez la taille pour faire face aux pics de trafic prévus : en cas de forte charge, le moteur peut ne plus répondre, ce qui affecte la disponibilité. `BytesUsedForCacheet` `DatabaseMemoryUsagePercentage` sont de bons indicateurs de votre utilisation de la mémoire, alors `ReplicationLag` qu'ils indiquent l'état de votre réplication en fonction de votre taux d'écriture. Vous pouvez utiliser ces métriques pour déclencher la mise à l'échelle du cluster.
6. Assurez la résilience côté client en effectuant des tests à l'aide de l'API Failover avant un événement de basculement de production.

[Ressources] :

- Configuration ElastiCache pour Redis OSS pour une meilleure disponibilité
- Haute disponibilité avec les groupes de réplication

REL 2 : Comment atteignez-vous vos objectifs de point de reprise (RPOs) ElastiCache ?

Introduction au niveau des questions : comprenez le RPO de la charge de travail pour éclairer les décisions relatives aux stratégies de ElastiCache sauvegarde et de restauration.

Avantage au niveau de la question : la mise en place d'une stratégie de RPO peut améliorer la continuité des activités en cas de scénario de reprise après sinistre. La conception de vos politiques de sauvegarde et de restauration peut vous aider à atteindre vos objectifs de point de restauration (RPO) pour vos ElastiCache données. ElastiCache propose des fonctionnalités de capture instantanée stockées dans Amazon S3, ainsi qu'une politique de conservation configurable. Ces instantanés sont pris au cours d'une fenêtre de sauvegarde définie et sont gérés automatiquement par le service. Si votre charge de travail nécessite une granularité de sauvegarde supplémentaire, vous avez la possibilité de créer jusqu'à 20 sauvegardes manuelles par jour. Les sauvegardes créées manuellement ne sont pas soumises à une politique de conservation de service et peuvent être conservées indéfiniment.

- [Obligatoire] Comprenez et documentez le RPO de vos ElastiCache déploiements.
  - Sachez que Memcached ne propose aucun processus de sauvegarde.
  - Passez en revue les fonctionnalités de ElastiCache Backup and Restore.
- [Meilleure pratique] Mettez en place un processus bien communiqué pour la sauvegarde de votre cluster.
  - Lancez des sauvegardes manuelles selon vos besoins.
  - Passez en revue les politiques de conservation pour les sauvegardes automatiques.
  - Notez que les sauvegardes manuelles seront conservées indéfiniment.
  - Planifiez vos sauvegardes automatiques pendant les périodes de faible utilisation.
  - Effectuez des opérations de sauvegarde sur des réplicas en lecture afin de minimiser l'impact sur les performances du cluster.
- [Bien] Tirez parti de la fonction de sauvegarde planifiée ElastiCache pour sauvegarder régulièrement vos données pendant une période définie.
  - Testez régulièrement les restaurations à partir de vos sauvegardes.
- [Ressources] :
  - Système d'exploitation Redis
  - Backup et restauration pour ElastiCache
  - Réalisation de sauvegardes manuelles

- Planification des sauvegardes automatiques
- ElastiCache Clusters de sauvegarde et de restauration

PF 3 : Comment répondez-vous aux exigences de reprise après sinistre (DR) ?

Introduction au niveau des questions : La reprise après sinistre est un aspect important de toute planification de la charge de travail. ElastiCache propose plusieurs options pour mettre en œuvre la reprise après sinistre en fonction des exigences de résilience de la charge de travail. Avec Amazon ElastiCache Global Datastore, vous pouvez écrire dans votre cluster situé dans une région et disposer des données disponibles pour être lues à partir de deux autres clusters de répliques interrégionaux, ce qui permet des lectures à faible latence et une reprise après sinistre dans toutes les régions.

Avantage au niveau de la question : la compréhension et la planification de divers scénarios de sinistre peuvent garantir la continuité des activités. Les stratégies de reprise après sinistre doivent être équilibrées en termes de coût, d'impact sur les performances et de risque de perte de données.

- [Obligatoire] Développez et documentez des stratégies de reprise après sinistre pour tous vos ElastiCache composants en fonction des exigences de charge de travail. ElastiCache est unique en ce sens que certains cas d'utilisation sont totalement éphémères et ne nécessitent aucune stratégie de reprise après sinistre, tandis que d'autres se situent à l'opposé et nécessitent une stratégie de reprise après sinistre extrêmement robuste. Toutes les options doivent être évaluées par rapport à l'optimisation des coûts : une meilleure résilience exige de plus grandes quantités d'infrastructure.

  Découvrez les options de reprise après sinistre disponibles au niveau régional et multirégional.

  - Les déploiements multi-AZ sont recommandés pour se prémunir contre les pannes de zone de disponibilité. Assurez-vous de déployer avec le mode cluster activé dans les architectures multi-AZ, avec un minimum de 3 disponibles. AZs
  - L'entrepôt de données global est recommandé pour se prémunir contre les défaillances régionales.
- [Meilleure pratique] Activez l'entrepôt de données global pour les charges de travail qui nécessitent une résilience au niveau de la région.
  - Prévoyez un basculement vers la région secondaire en cas de dégradation du cluster principal.
  - Testez le processus de basculement multirégional avant un basculement en production.

  - Surveillez la métrique `ReplicationLag` pour comprendre l'impact potentiel de la perte de données lors des événements de basculement.
- [Ressources] :
  - Atténuation des défaillances
  - Réplication entre AWS régions à l'aide de banques de données mondiales
  - Restauration à partir d'une sauvegarde avec redimensionnement facultatif du cluster
  - Minimiser les temps d'arrêt ElastiCache pour Valkey et Redis OSS avec Multi-AZ

PF 4 : Comment planifiez-vous efficacement les basculements ?

Introduction au niveau des questions : L'activation du multi-AZ avec des basculements automatiques est une bonne pratique. ElastiCache Dans certains cas, ElastiCache pour Valkey et Redis, OSS remplace les nœuds principaux dans le cadre des opérations de service. Par exemple, lors d'événements de maintenance planifiée et dans le cas improbable d'une défaillance du nœud ou d'un problème avec la zone de disponibilité. La réussite d'un basculement dépend à la fois de la configuration de votre bibliothèque cliente ElastiCache et de votre bibliothèque cliente.

Avantage au niveau des questions : le respect des meilleures pratiques en matière de ElastiCache basculement en conjonction avec votre bibliothèque ElastiCache client spécifique vous permet de minimiser les temps d'arrêt potentiels lors d'événements de basculement.

- [Obligatoire] Lorsque le mode cluster est désactivé, utilisez les délais d'expiration afin que vos clients détectent s'ils doivent se déconnecter de l'ancien nœud primaire et se reconnecter au nouveau nœud primaire, à l'aide de l'adresse IP du point de terminaison principal mise à jour. Lorsque le mode cluster est activé, la bibliothèque cliente est chargée de détecter les modifications de la topologie du cluster sous-jacent. Cela se fait le plus souvent par le biais des paramètres de configuration de la bibliothèque ElastiCache cliente, qui vous permettent également de configurer la fréquence et la méthode d'actualisation. Chaque bibliothèque cliente propose ses propres paramètres et des détails supplémentaires sont disponibles dans la documentation correspondante.

  [Ressources] :
  - Minimiser les temps d'arrêt ElastiCache pour Valkey et Redis OSS avec Multi-AZ
  - Passez en revue les meilleures pratiques de votre bibliothèque ElastiCache cliente.
- [Obligatoire] La réussite des basculements dépend de l'intégrité de l'environnement de réplication entre le nœud primaire et le nœud de réplica. Examinez et comprenez la nature asynchrone de

la réplication Valkey et Redis OSS, ainsi que les CloudWatch mesures disponibles pour signaler le délai de réplication entre les nœuds principaux et les nœuds de réplication. Pour les cas d'utilisation nécessitant une plus grande sécurité des données, utilisez la commande WAIT pour forcer les répliques à accuser réception des écritures avant de répondre aux clients connectés.

[Ressources] :

- Métriques pour Valkey ou Redis OSS
- Surveillance des meilleures pratiques liées à ElastiCache l'utilisation d'Amazon CloudWatch

- [Mieux] Validez régulièrement la réactivité de votre application lors du basculement à l'aide de l'API ElastiCache Test Failover.

  [Ressources] :

  - Test du basculement automatique vers une réplique en lecture sur ElastiCache
  - Test du basculement automatique

REL 5 : Vos ElastiCache composants sont-ils conçus pour évoluer ?

Introduction au niveau des questions : en comprenant les capacités de mise à l'échelle et les topologies de déploiement disponibles, vos ElastiCache composants peuvent s'adapter au fil du temps pour répondre à l'évolution des exigences de charge de travail. ElastiCachepropose une mise à l'échelle à 4 voies : in/out (horizontale) et up/down (verticale).

Avantage au niveau des questions : le respect des meilleures pratiques en matière de ElastiCache déploiement garantit la plus grande flexibilité de mise à l'échelle, tout en respectant le principe Well Architected qui consiste à effectuer une mise à l'échelle horizontale afin de minimiser l'impact des défaillances.

- [Obligatoire] Comprenez la différence entre les topologies « mode cluster activé » et « mode cluster désactivé ». Dans presque tous les cas, il est recommandé d'effectuer un déploiement en ayant activé le mode Cluster, car il permet de renforcer la capacité de mise à l'échelle au fil du temps. Les composants pour lesquels le mode cluster est désactivé sont limités dans leur capacité à être mis à l'échelle horizontalement en ajoutant des réplicas en lecture.
- [Obligatoire] Sachez quand et comment procéder à une mise à l'échelle.
  - Pour augmenter le nombre de READIOPS : ajoutez des réplicas.
  - Pour augmenter le nombre de WRITEOPS : ajoutez des partitions (montée en puissance).

  - Pour augmenter le nombre d'E/S sur le réseau : utilisez des instances optimisées pour le réseau, augmentez la capacité.
- [Mieux] Déployez vos ElastiCache composants en activant le mode cluster, en privilégiant un plus grand nombre de nœuds plus petits plutôt qu'un nombre réduit de nœuds plus grands. Cela limite efficacement le rayon d'explosion d'une défaillance de nœud.
- [Meilleure pratique] Incluez des réplicas dans vos clusters pour améliorer la réactivité lors des événements de mise à l'échelle.
- [Bien] Si le mode cluster est désactivé, utilisez les répliques de lecture pour augmenter la capacité de lecture globale. ElastiCache prend en charge jusqu'à 5 répliques de lecture en mode cluster désactivée, ainsi que la mise à l'échelle verticale.
- [Ressources] :
  - Dimensionnement ElastiCache des clusters
  - Augmentation en ligne

Pilier d'efficacité des ElastiCache performances des objectifs Amazon Well-Architected

Le pilier Efficacité des performances met l'accent sur l'utilisation efficace des ressources informatiques et de calcul. Les sujets clés incluent la sélection des types et des tailles de ressources appropriés en fonction des exigences de charge de travail, la surveillance des performances et la prise de décisions éclairées pour maintenir l'efficacité à mesure que les besoins de l'entreprise évoluent.

Rubriques

- PE 1 : Comment surveillez-vous les performances de votre ElastiCache cluster Amazon ?
- PE 2 : Comment répartissez-vous le travail entre les nœuds de votre ElastiCache cluster ?
- EP 3 : Pour la mise en cache des charges de travail, comment suivez-vous et rendez-vous compte de l'efficacité et des performances de votre cache ?
- EP 4 : Comment votre charge de travail optimise-t-elle l'utilisation des ressources de mise en réseau et des connexions ?
- PE 5 : Comment gérez-vous l' and/or expulsion par suppression de clés ?
- PE 6 : Comment modélisez-vous les données et interagissez-vous avec celles-ci ElastiCache ?
- PE 7 : Comment enregistrez-vous les commandes lentes dans votre ElastiCache cluster Amazon ?

- PE8: Comment Auto Scaling contribue-t-il à améliorer les performances du ElastiCache cluster ?

PE 1 : Comment surveillez-vous les performances de votre ElastiCache cluster Amazon ?

Introduction au niveau de la question : en comprenant les métriques de surveillance existantes, vous pouvez identifier l'utilisation actuelle. Une surveillance appropriée peut aider à identifier les goulots d'étranglement potentiels ayant une incidence sur les performances de votre cluster.

Avantage au niveau de la question : la compréhension des métriques associées à votre cluster peut aider à orienter les techniques d'optimisation susceptibles de réduire la latence et d'augmenter le débit.

- [Obligatoire] Tests de performance de base en utilisant un sous-ensemble de votre charge de travail.
  - Vous devez surveiller les performances de la charge de travail réelle à l'aide de mécanismes tels que les tests de charge.
  - Surveillez les CloudWatch indicateurs lors de l'exécution de ces tests afin de comprendre les indicateurs disponibles et d'établir une référence de performance.
- [Idéal] ElastiCache Pour les charges de travail Valkey et Redis OSS, renommez les commandes coûteuses en termes de calcul, par exemple pour limiter la capacité des utilisateurs à exécuter des commandes de blocage sur des clusters de production. `KEYS`
  - ElastiCache les charges de travail exécutant le moteur 6.x pour Redis OSS peuvent tirer parti du contrôle d'accès basé sur les rôles pour restreindre certaines commandes. L'accès aux commandes peut être contrôlé en créant des utilisateurs et des groupes d'utilisateurs à l'aide de la AWS console ou de la CLI, et en associant les groupes d'utilisateurs à un cluster. Dans Redis OSS 6, lorsque le RBAC est activé, nous pouvons utiliser « - @dangerous » et cela interdira les commandes coûteuses telles que KEYS, MONITOR, SORT, etc. pour cet utilisateur.
  - Pour la version 5.x du moteur, renommez les commandes à l'aide du `rename-commands` paramètre du groupe de paramètres du cluster.
- [Pratique encore meilleure] Analysez les requêtes lentes et recherchez des techniques d'optimisation.
  - ElastiCache Pour les charges de travail Valkey et Redis OSS, apprenez-en plus sur vos requêtes en analysant le journal lent. Par exemple, vous pouvez utiliser la commande `valkey-cli slowlog get 10` pour afficher les 10 dernières commandes qui ont dépassé les seuils de latence (10 secondes par défaut).

- Certaines requêtes peuvent être effectuées plus efficacement à l'aide de structures de données complexes ElastiCache pour Valkey et Redis OSS. Par exemple, pour les consultations de plages de style numérique, une application peut implémenter des index numériques simples avec des ensembles triés. La gestion de ces index permet de réduire les analyses effectuées sur l'ensemble de données et de renvoyer les données avec une meilleure efficacité en termes de performances.
- ElastiCache Pour les charges de travail Valkey et Redis OSS, `redis-benchmark` fournit une interface simple pour tester les performances de différentes commandes à l'aide d'entrées définies par l'utilisateur, telles que le nombre de clients et la taille des données.
- Étant donné que Memcached ne prend en charge que les commandes simples au niveau de la clé, pensez à créer des clés supplémentaires sous forme d'index afin d'éviter d'itérer dans l'espace de clé pour répondre aux requêtes des clients.

- [Ressources] :
  - Surveillance de l'utilisation à l'aide de CloudWatch métriques
  - Utilisation des CloudWatch alarmes Amazon
  - Paramètres spécifiques à Valkey et Redis OSS
  - SLOWLOG
  - point de référence

PE 2 : Comment répartissez-vous le travail entre les nœuds de votre ElastiCache cluster ?

Introduction au niveau des questions : la façon dont votre application se connecte ElastiCache aux nœuds Amazon peut avoir un impact sur les performances et l'évolutivité du cluster.

Avantage au niveau de la question : l'utilisation appropriée des nœuds disponibles dans le cluster garantira la répartition du travail entre les ressources disponibles. Les techniques suivantes permettent également d'éviter les ressources inactives.

- [Obligatoire] Demandez aux clients de se connecter au point de ElastiCache terminaison approprié.
  - ElastiCache pour Valkey et Redis OSS implémente différents points de terminaison en fonction du mode cluster utilisé. Si le mode cluster est activé, ElastiCache fournira un point de terminaison de configuration. Lorsque le mode cluster est désactivé, ElastiCache fournit un point de terminaison principal, généralement utilisé pour les écritures, et un point de terminaison de lecteur pour équilibrer les lectures entre les répliques. L'implémentation correcte de ces points

de terminaison se traduit par de meilleures performances et des opérations de mise à l'échelle simplifiées. Évitez de vous connecter à des points de terminaison de nœuds individuels, sauf exigences particulières.

- Pour les clusters Memcached à nœuds multiples, ElastiCache fournit un point de terminaison de configuration qui active la découverte automatique. Il est recommandé d'utiliser un algorithme de hachage pour répartir le travail de manière uniforme entre les nœuds de cache. De nombreuses bibliothèques clientes Memcached implémentent un hachage cohérent. Consultez la documentation de la bibliothèque que vous utilisez pour voir si elle prend en charge le hachage cohérent et comment le mettre en œuvre. Vous trouverez plus d'informations sur l'implémentation de ces fonctionnalités ici.

- [Mieux] Tirez parti des clusters compatibles avec le mode cluster ElastiCache pour Valkey et Redis OSS pour améliorer l'évolutivité.
  - ElastiCache pour Valkey et Redis OSS (mode cluster activé), les clusters prennent en charge les opérations de dimensionnement en ligne (out/in and up/down) pour aider à distribuer les données de manière dynamique entre les partitions. L'utilisation du point de terminaison de configuration permet à vos clients qui connaissent le cluster de s'adapter aux modifications de la topologie du cluster.
  - Vous pouvez également rééquilibrer le cluster en déplaçant les hashslots entre les partitions disponibles dans votre cluster ElastiCache for Valkey et Redis OSS (mode cluster activé). Vous pourrez ainsi répartir le travail de manière plus efficace entre les partitions disponibles.
- [Pratique encore meilleure] Implémentez une stratégie pour identifier et corriger les touches de raccourci de votre charge de travail.
  - Tenez compte de l'impact des structures de données multidimensionnelles Valkey ou Redis OSS telles que les listes, les flux, les ensembles, etc. Ces structures de données sont stockées dans des clés uniques, qui résident sur un seul nœud. Une clé multidimensionnelle très volumineuse est susceptible d'utiliser davantage de capacité réseau et de mémoire que les autres types de données et peut entraîner une utilisation disproportionnée de ce nœud. Si possible, concevez votre charge de travail de manière à répartir l'accès aux données sur de nombreuses clés discrètes.
  - Les touches de raccourci de la charge de travail peuvent avoir un impact sur les performances du nœud utilisé. ElastiCache Pour les charges de travail Valkey et Redis OSS, vous pouvez détecter les touches de raccourci en utilisant `valkey-cli --hotkeys` si une politique de mémoire maximale LFU est en place.
  - Envisagez de répliquer les touches de raccourci sur plusieurs nœuds afin de répartir leur accès de manière plus uniforme. Cette approche oblige le client à écrire sur plusieurs nœuds

principaux (le nœud Valkey ou Redis OSS lui-même ne fournira pas cette fonctionnalité) et à conserver une liste de noms de clés à lire, en plus du nom de clé d'origine.

- ElastiCache le moteur 7.2 pour Valkey et versions ultérieures, et la ElastiCache version 6 pour Redis OSS et versions ultérieures, prennent tous en charge la mise en cache côté client assistée par le serveur. Cela permet aux applications d'attendre que des modifications soient apportées à une clé avant de renvoyer des appels réseau àElastiCache.

- [Ressources] :
  - Configuration ElastiCache pour Valkey et Redis OSS pour une meilleure disponibilité
  - Recherche de points de terminaison de connexion dans ElastiCache
  - Bonnes pratiques en matière d'équilibrage de charge
  - Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)
  - Mise en cache côté client dans Valkey et Redis OSS

EP 3 : Pour la mise en cache des charges de travail, comment suivez-vous et rendez-vous compte de l'efficacité et des performances de votre cache ?

Introduction au niveau des questions : La mise en cache est une charge de travail courante ElastiCache et il est important que vous sachiez comment gérer l'efficacité et les performances de votre cache.

Avantage au niveau de la question : votre application peut montrer des signes de lenteur des performances. Votre capacité à utiliser des métriques spécifiques au cache pour prendre des décisions éclairées quant à la manière d'améliorer les performances des applications est essentielle pour la charge de travail de votre cache.

- [Obligatoire] Mesurez et suivez au fil du temps le taux d'accès au cache. L'efficacité de votre cache est déterminée par son « taux d'accès au cache ». Le taux d'accès au cache est défini par le nombre total d'accès à une clé divisé par le nombre total d'accès et d'échecs. Plus le taux est proche de 1, plus votre cache est efficace. Un faible taux d'accès au cache est dû au volume d'échecs d'accès au cache. Les échecs d'accès au cache se produisent lorsque la clé demandée n'est pas trouvée dans le cache. Une clé est introuvable dans le cache, car elle a été expulsée ou supprimée, a expiré ou n'a jamais existé. Déterminez pourquoi les clés ne figurent pas dans le cache et développez des stratégies appropriées pour les conserver dans le cache.

  [Ressources] :

  - Métriques pour Valkey et Redis OSS

- [Obligatoire] Mesurez et collectez les performances du cache de votre application en conjonction avec les valeurs de latence et d'utilisation du processeur afin de déterminer si vous devez apporter des modifications à vos composants time-to-live ou à d'autres composants de l'application. ElastiCache fournit un ensemble de CloudWatch mesures pour les latences agrégées pour chaque structure de données. Ces mesures de latence sont calculées à l'aide des statistiques commandstats de la commande INFO et n'incluent ni le réseau ni le temps d'E/S. Il s'agit uniquement du temps nécessaire ElastiCache au traitement des opérations.

  [Ressources] :

  - Métriques pour Valkey et Redis OSS
  - Surveillance des meilleures pratiques liées à ElastiCache l'utilisation d'Amazon CloudWatch
- [Meilleure pratique] Choisissez la stratégie de mise en cache adaptée à vos besoins. Un faible taux d'accès au cache est dû au volume d'échecs d'accès au cache. Si votre charge de travail est conçue pour présenter un faible volume d'échecs d'accès au cache (telles que les communications en temps réel), il est préférable de passer en revue vos stratégies de mise en cache et d'appliquer les résolutions les plus appropriées à votre charge de travail, telles que l'instrumentation des requêtes pour mesurer la mémoire et les performances. Les stratégies réelles que vous mettez en œuvre pour remplir et assurer la maintenance de votre cache dépendent des données que vos clients ont besoin de mettre en cache et des modèles d'accès à ces données. Par exemple, il est peu probable que vous utilisiez la même stratégie à la fois pour les recommandations personnalisées sur une application de streaming et pour les actualités tendances.

  [Ressources] :

  - Stratégies de mise en cache pour Memcached
  - Caching Best Practices
  - Des performances à grande échelle avec le ElastiCache livre blanc Amazon

EP 4 : Comment votre charge de travail optimise-t-elle l'utilisation des ressources de mise en réseau et des connexions ?

Introduction au niveau des questions : ElastiCache pour Valkey, Memcached et Redis, les logiciels OSS sont pris en charge par de nombreux clients d'applications, et les implémentations peuvent varier. Vous devez comprendre la gestion de la mise en réseau et des connexions en place pour analyser l'impact potentiel sur les performances.

Avantage au niveau de la question : une utilisation efficace des ressources de mise en réseau peut améliorer l'efficacité des performances de votre cluster. Les recommandations suivantes peuvent réduire les demandes de mise en réseau et améliorer la latence et le débit du cluster.

- [Obligatoire] Gérez les connexions à votre ElastiCache cluster de manière proactive.
  - Le regroupement des connexions dans l'application réduit la surcharge sur le cluster créée par l'ouverture et la fermeture de connexions. Surveillez le comportement de connexion sur Amazon à CloudWatch l'aide `CurrConnections` de et`NewConnections`.
  - Évitez les fuites de connexion en fermant correctement les connexions client, le cas échéant. Les stratégies de gestion des connexions consistent notamment à fermer correctement les connexions qui ne sont pas utilisées et à définir des délais d'expiration de connexion.
  - Pour les charges de travail Memcached, il existe une quantité configurable de mémoire réservée à la gestion des connexions appelée `memcached_connections_overhead`.
- [Pratique encore meilleure] Compressez les objets volumineux pour réduire la mémoire et améliorer le débit du réseau.
  - La compression des données peut réduire le débit réseau requis (Gbit/s), mais augmente la charge de travail de l'application pour compresser et décompresser les données.
  - La compression réduit également la quantité de mémoire consommée par les touches.
  - En fonction des besoins de votre application, trouvez le juste équilibre entre le taux de compression et la vitesse de compression.
- [Ressources] :
  - ElastiCache - Banque de données mondiale
  - Paramètres spécifiques Memcached
  - ElastiCache la version 5.0.3 pour Redis OSS améliore la I/O manipulation pour améliorer les performances
  - Métriques pour Valkey et Redis OSS
  - Configuration ElastiCache pour une meilleure disponibilité

PE 5 : Comment gérez-vous l' and/or expulsion par suppression de clés ?

Introduction au niveau des questions : les charges de travail ont des exigences et un comportement attendu différents lorsqu'un nœud de cluster approche des limites de consommation de mémoire. ElastiCache a des politiques différentes pour gérer ces situations.

Avantage au niveau de la question : une gestion appropriée de la mémoire disponible et la compréhension des politiques d'expulsion permettent de prendre conscience du comportement du cluster lorsque les limites de mémoire de l'instance sont dépassées.

- [Obligatoire] Instrumentez l'accès aux données pour déterminer la politique à appliquer. Identifiez une politique de mémoire maximale appropriée pour contrôler si et comment les expulsions sont effectuées sur le cluster.
  - L'expulsion se produit lorsque la mémoire maximale du cluster est consommée et qu'une politique est en place pour autoriser l'expulsion. Le comportement du cluster dans cette situation dépend de la politique d'expulsion spécifiée. Cette politique peut être gérée à l'aide du groupe de paramètres `maxmemory-policy` sur le cluster.
  - La politique par défaut `volatile-lru` libère de la mémoire en expulsant les clés dont le délai d'expiration est défini (valeur TTL). Les politiques les moins fréquemment utilisées (LFU) et les moins récemment utilisées (LRU) suppriment les clés en fonction de leur utilisation.
  - Pour les charges de travail Memcached, une politique LRU par défaut a été mise en place pour contrôler les expulsions sur chaque nœud. Le nombre d'expulsions sur votre ElastiCache cluster Amazon peut être surveillé à l'aide de la métrique Expulsions sur Amazon. CloudWatch
- [Pratique encore meilleure] Normalisez le comportement de suppression de sorte à contrôler l'impact sur les performances de votre cluster afin d'éviter les goulets d'étranglement inattendus en matière de performances.
  - ElastiCache Pour les charges de travail Valkey et Redis OSS, lorsque vous supprimez explicitement des clés du cluster, cela `UNLINK` revient à supprimer les `DEL` clés spécifiées. Cependant, la commande effectue la récupération de mémoire proprement dite dans un thread différent, de sorte qu'elle ne bloque pas, contrairement à `DEL`. La suppression proprement dite se fera ultérieurement de manière asynchrone.
  - Pour les charges de travail de la ElastiCache version 6.x pour Redis OSS, le comportement de la `DEL` commande peut être modifié dans le groupe de paramètres à l'aide du paramètre. `lazyfree-lazy-user-del`
- [Ressources] :
  - Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres
  - UNLINK
  - Gestion financière dans le cloud avec AWS

PE 6 : Comment modélisez-vous les données et interagissez-vous avec celles-ci ElastiCache ?

Introduction au niveau des questions : ElastiCache dépend fortement de l'application des structures de données et du modèle de données utilisés, mais elle doit également prendre en compte le magasin de données sous-jacent (le cas échéant). Comprenez les structures de données disponibles et assurez-vous d'utiliser les structures de données les mieux adaptées à vos besoins.

Avantage au niveau des questions : la modélisation des données ElastiCache comporte plusieurs couches, notamment les cas d'utilisation des applications, les types de données et les relations entre les éléments de données. De plus, chaque type de données et chaque commande ont leurs propres signatures de performance bien documentées.

- [Meilleure pratique] L'une des meilleures pratiques consiste à réduire la réécriture involontaire de données. Utilisez une convention de dénomination qui minimise le chevauchement des noms de clé. La dénomination conventionnelle de vos structures de données utilise une méthode hiérarchique telle que : `APPNAME:CONTEXT:ID`, par exemple : `ORDER-APP:CUSTOMER:123`.

  [Ressources] :

  - Key naming
- [Idéal] ElastiCache pour Valkey et Redis, les commandes OSS ont une complexité temporelle définie par la notation Big O. Cette fois, la complexité d'une commande est une algorithmic/mathematical représentation de son impact. Lorsque vous introduisez un nouveau type de données dans votre application, vous devez examiner attentivement la complexité temporelle des commandes associées. Les commandes dont la complexité temporelle est O(1) sont constantes dans le temps et ne dépendent pas de la taille de l'entrée. Toutefois, les commandes dont la complexité temporelle est O(N) sont linéaires dans le temps et dépendent de la taille de l'entrée. En raison de la conception à thread unique de ElastiCache Valkey et Redis OSS, un volume important d'opérations complexes dans le temps entraînera une baisse des performances et des délais d'exécution potentiels.

  [Ressources] :

  - Commandes
- [Idéal] APIs À utiliser pour obtenir une visibilité de l'interface graphique sur le modèle de données de votre cluster.

  [Ressources] :

- Commandant Redis OSS
- Navigateur Redis OSS
- Redsmin

PE 7 : Comment enregistrez-vous les commandes lentes dans votre ElastiCache cluster Amazon ?

Introduction au niveau de la question : l'optimisation des performances bénéficie à la capture, l'agrégation et la notification des commandes de longue durée. En comprenant le temps nécessaire à l'exécution des commandes, vous pouvez déterminer les commandes qui nuisent aux performances et celles qui empêchent le moteur de fonctionner de manière optimale. ElastiCache a également la capacité de transmettre ces informations à Amazon CloudWatch ou Amazon Kinesis Data Firehose.

Avantage au niveau de la question : la journalisation dans un emplacement permanent dédié et la fourniture d'événements de notification pour les commandes lentes peuvent faciliter l'analyse détaillée des performances et peuvent être utilisées pour déclencher des événements automatisés.

- [Obligatoire] ElastiCache exécution d'un moteur Valkey 7.2 ou version ultérieure, ou exécution d'un moteur Redis OSS version 6.0 ou ultérieure, groupe de paramètres correctement configuré et journalisation SLOWLOG activée sur le cluster.
  - Les paramètres requis ne sont disponibles que lorsque la compatibilité des versions du moteur est définie sur Valkey 7.2 ou version ultérieure, ou sur Redis OSS version 6.0 ou supérieure.
  - La journalisation SLOWLOG se produit lorsque le temps d'exécution d'une commande par le serveur dépasse la valeur spécifiée. Le comportement du cluster dépend des paramètres du groupe de paramètres associés, qui sont `slowlog-log-slower-than` et `slowlog-max-len`.
  - Les modifications prennent effet immédiatement.
- [Mieux] Tirez parti des fonctionnalités de CloudWatch Kinesis Data Firehose.
  - Utilisez les fonctionnalités de filtrage et d'alarme de CloudWatch CloudWatch Logs Insights et Amazon Simple Notification Services pour surveiller les performances et notifier les événements.
  - Utilisez les fonctionnalités de streaming de Kinesis Data Firehose pour archiver les journaux SLOWLOG dans un stockage permanent ou pour déclencher le réglage automatique des paramètres du cluster.
  - Déterminez quel format entre JSON et TEXT brut répond le mieux à vos besoins.
  - Fournissez les autorisations IAM pour publier sur Kinesis Data Firehose CloudWatch ou sur Kinesis Data Firehose.

- [Pratique encore meilleure] Définissez `slowlog-log-slower-than` sur une valeur autre que la valeur par défaut.
  - Ce paramètre détermine la durée pendant laquelle une commande peut être exécutée dans le moteur Valkey ou Redis OSS avant d'être enregistrée en tant que commande lente. La valeur par défaut est 10 000 microsecondes (10 millisecondes). La valeur par défaut est peut-être trop élevée pour certaines charges de travail.
  - Déterminez une valeur plus adaptée à votre charge de travail en fonction des besoins de l'application et des résultats des tests ; toutefois, une valeur trop faible peut générer un volume de données excessif.
- [Pratique encore meilleure] Conservez `slowlog-max-len` comme valeur par défaut.
  - Ce paramètre détermine la limite supérieure du nombre de commandes lentes capturées dans la mémoire Valkey ou Redis OSS à un moment donné. La valeur 0 désactive effectivement la capture. Plus la valeur est élevée, plus le nombre d'entrées stockées en mémoire est élevé, ce qui réduit le risque d'expulsion d'informations importantes avant qu'elles aient pu être consultées. La valeur par défaut est 128.
  - La valeur par défaut convient à la plupart des charges de travail. S'il est nécessaire d'analyser les données dans une fenêtre temporelle étendue depuis le valkey-cli via la commande SLOWLOG, envisagez d'augmenter cette valeur. Cela permet à davantage de commandes de rester dans la mémoire de Valkey ou Redis OSS.

    Si vous émettez les données SLOWLOG vers CloudWatch Logs ou Kinesis Data Firehose, les données seront conservées et pourront être analysées en dehors du système, ce qui réduit ElastiCache le besoin de stocker un grand nombre de commandes lentes dans la mémoire Valkey ou Redis OSS.
- [Ressources] :
  - Comment activer la journalisation lente dans un cluster de cache ?
  - Livraison de journaux
  - Paramètres spécifiques à Redis OS
  - https://aws.amazon.com/cloudwatch/Amazon CloudWatch
  - Amazon Kinesis Data Firehose

PE8: Comment Auto Scaling contribue-t-il à améliorer les performances du ElastiCache cluster ?

Introduction au niveau des questions : en implémentant la fonctionnalité de mise à l'échelle automatique de Valkey ou Redis OSS, ElastiCache vos composants peuvent s'ajuster au fil du temps pour augmenter ou diminuer automatiquement les fragments ou les répliques souhaités. Pour ce faire, vous pouvez implémenter le suivi des cibles ou une politique de mise à l'échelle planifiée.

Avantage au niveau des questions : la compréhension et la planification des pics de charge de travail peuvent garantir des performances de mise en cache améliorées et la continuité des activités. ElastiCache Auto Scaling surveille en permanence l'utilisation de votre processeur et de votre mémoire pour s'assurer que votre cluster fonctionne aux niveaux de performance souhaités.

- [Obligatoire] Lors du lancement d'un cluster ElastiCache pour Valkey ou Redis OSS :
  1. Assurez-vous que le mode Cluster est activé.
  2. Assurez-vous que l'instance appartient à une famille d'un certain type et d'une certaine taille qui prend en charge l'autoscaling.
  3. Assurez-vous que le cluster ne s'exécute pas dans des entrepôts de données globaux, des outposts ou des zones locales.

  [Ressources] :
  - Mise à l'échelle des clusters dans Valkey et Redis OSS (mode cluster activé)
  - Utilisation de Auto Scaling avec des partitions
  - Utilisation d'Auto Scaling avec des réplicas
- [Meilleure pratique] Déterminez si votre charge de travail est à lecture ou à écriture intensive pour définir une politique de mise à l'échelle. Pour de meilleures performances, utilisez une seule métrique de suivi. Il est recommandé de ne pas appliquer plusieurs politiques à chaque dimension. En effet, les stratégies d'autoscaling montent en puissance lorsque la cible est atteinte, mais ne sont mises à l'échelle horizontale que lorsque toutes les politiques de suivi des cibles sont prêtes à être mises à l'échelle horizontale.

  [Ressources] :
  - Politiques Auto Scaling
  - Définition d'une politique de mise à l'échelle
- [Meilleure pratique] La surveillance des performances au fil du temps peut vous aider à détecter les modifications de la charge de travail qui passeraient inaperçues si elles étaient surveillées à

un moment donné. Vous pouvez analyser les CloudWatch mesures correspondantes relatives à l'utilisation du cluster sur une période de quatre semaines afin de déterminer le seuil de valeur cible. Si vous n'êtes toujours pas sûr de la valeur à choisir, nous vous recommandons de commencer par une valeur de métrique prédéfinie minimale prise en charge.

[Ressources] :

- Surveillance de l'utilisation à l'aide de CloudWatch métriques

- [Mieux] Nous vous conseillons de tester votre application avec les charges de travail minimales et maximales attendues, afin d'identifier le nombre exact de charges shards/replicas requises pour que le cluster développe des politiques de dimensionnement et atténue les problèmes de disponibilité.

  [Ressources] :

  - Enregistrement d'une cible évolutive
  - Enregistrement d'une cible évolutive à l'aide du AWS CLI

Pilier d'optimisation des ElastiCache coûts d'objectif Amazon Well-Architected

Le pilier d'optimisation des coûts vise à éviter les coûts inutiles. Les sujets clés incluent la compréhension et le contrôle des dépenses, la sélection du type de nœud le plus approprié (utilisation d'instances qui prennent en charge la hiérarchisation des données en fonction des besoins de la charge de travail), le nombre approprié de types de ressources (nombre de réplicas en lecture), l'analyse des dépenses au fil du temps et la mise à l'échelle pour répondre aux besoins de l'entreprise sans dépenses excessives.

Rubriques

- COÛT 1 : Comment identifiez-vous et suivez-vous les coûts associés à vos ElastiCache ressources ? Comment développez-vous des mécanismes permettant aux utilisateurs de créer, gérer et supprimer les ressources créées ?
- COÛT 2 : Comment utilisez-vous les outils de surveillance continue pour vous aider à optimiser les coûts associés à vos ElastiCache ressources ?
- POC 3 : Devez-vous utiliser un type d'instance prenant en charge la hiérarchisation des données ? Quels sont les avantages de la hiérarchisation des données ? Quand ne pas utiliser les instances de hiérarchisation des données ?

COÛT 1 : Comment identifiez-vous et suivez-vous les coûts associés à vos ElastiCache ressources ? Comment développez-vous des mécanismes permettant aux utilisateurs de créer, gérer et supprimer les ressources créées ?

Introduction au niveau de la question : la compréhension des métriques de coût nécessite la participation et la collaboration de plusieurs équipes : ingénierie logicielle, gestion des données, propriétaires de produits, finances et direction. Pour identifier les principaux facteurs de coûts, toutes les parties concernées doivent comprendre les leviers de contrôle de l'utilisation des services et les compromis en matière de gestion des coûts. C'est ce qui différencie souvent les efforts d'optimisation des coûts réussis et moins réussis. En vous assurant que vous disposez de processus et d'outils pour suivre les ressources créées, du développement à la production et au retrait, vous pouvez gérer les coûts associés àElastiCache.

Avantage au niveau des questions : le suivi continu de tous les coûts associés à votre charge de travail nécessite une compréhension approfondie de l'architecture qui inclut ElastiCache l'un de ses composants. En outre, vous devez disposer d'un plan de gestion des coûts pour collecter et comparer l'utilisation par rapport à votre budget.

- [Obligatoire] Créez un centre d'excellence cloud (CCoE) doté de l'une de ses chartes fondatrices pour définir, suivre et prendre des mesures en fonction des indicateurs relatifs à l'utilisation de votre organisationElastiCache . Si un CCo E existe et fonctionne, assurez-vous qu'il sait comment lire et suivre les coûts associés ElastiCache. Lorsque des ressources sont créées, utilisez les rôles et les politiques IAM pour vérifier que seuls des équipes et des groupes spécifiques peuvent instancier des ressources. Cela garantit que les coûts sont associés aux résultats commerciaux et qu'une ligne de responsabilité claire est établie, du point de vue des coûts.
  1. CCoE doit identifier, définir et publier des indicateurs de coûts qui sont mis à jour régulièrement (mensuellement) en fonction de ElastiCache l'utilisation clé de données catégorielles telles que :
     a. Types de nœuds utilisés et leurs attributs : standard ou à mémoire optimisée, instances à la demande ou réservées, régions et zones de disponibilité
     b. Types d'environnements : libre, développement, test et production
     c. Stratégies de stockage et de conservation des sauvegardes
     d. Transfert de données au sein des régions et entre elles
     e. Instances s'exécutant sur Amazon Outposts
  2. CCoE se compose d'une équipe interfonctionnelle composée de représentants non exclusifs des équipes de génie logiciel, de gestion des données, de l'équipe produit, des finances et de direction de votre organisation.

[Ressources] :

- Create a Cloud Center of Excellence
- ElastiCacheTarifs Amazon

- [Obligatoire] Utilisez des balises de répartition des coûts pour suivre les coûts avec un faible niveau de granularité. Utilisez la gestion des AWS coûts pour visualiser, comprendre et gérer vos AWS coûts et votre utilisation au fil du temps.

  1. Utilisez des balises pour organiser vos ressources et des balises de répartition des coûts pour suivre vos AWS coûts de manière détaillée. Après avoir activé les balises de répartition des coûts, AWS utilise les balises de répartition des coûts pour organiser les coûts des ressources dans votre rapport de répartition des coûts, afin de faciliter la catégorisation et le suivi de vos AWS coûts. AWS fournit deux types de balises de répartition des coûts : les balises AWS générées et les balises définies par l'utilisateur. AWS définit, crée et applique les balises AWS générées pour vous, et vous définissez, créez et appliquez des balises définies par l'utilisateur. Vous devez activer les deux types d'étiquettes séparément pour qu'elles apparaissent dans Cost Management ou sur un rapport de répartition des coûts.
  2. Utilisez des balises de répartition des coûts pour organiser votre AWS facture afin de refléter votre propre structure de coûts. Lorsque vous ajoutez des balises de répartition des coûts à vos ressources sur Amazon ElastiCache, vous pouvez suivre les coûts en regroupant les dépenses sur vos factures par valeur d'étiquette de ressource. Vous devriez également envisager de combiner des balises pour suivre les coûts plus détaillés.

  [Ressources] :

  - Utilisation des balises AWS de répartition des coûts
  - Surveillance des coûts avec des balises de répartition des coûts
  - AWS Cost Explorer

- [Mieux] Connectez les ElastiCache coûts aux indicateurs qui concernent l'ensemble de l'organisation.

  1. Prenez en compte les métriques commerciales ainsi que les métriques opérationnelles telles que la latence. Quels concepts de votre modèle commercial sont compréhensibles pour tous les rôles ? Les métriques doivent être compréhensibles par le plus grand nombre de rôles possible au sein de l'organisation.
  2. Exemples : utilisateurs servis simultanément, latence maximale et moyenne par opération et par utilisateur, scores d'engagement des utilisateurs, taux de retour des utilisateursrates/week, session length/user, taux d'abandon, taux de réussite du cache et suivi des clés

[Ressources] :

- Surveillance de l'utilisation à l'aide de CloudWatch métriques

- [Bon] Maintenez une visibilité up-to-date architecturale et opérationnelle sur les indicateurs et les coûts sur l'ensemble de la charge de travail utilisée ElastiCache.
  1. Comprenez l'ensemble de votre écosystème de solutions, qui fait ElastiCache généralement partie d'un écosystème complet de AWS services dans leur ensemble technologique, qu'il s'agisse des clients, d'API Gateway, de Redshift ou des QuickSight outils de reporting (par exemple).
  2. Cartographiez les composants de votre solution (clients, connexions, sécurité, opérations en mémoire, stockage, automatisation des ressources, accès aux données et gestion des données) sur votre diagramme d'architecture. Chaque couche est connectée à l'ensemble de la solution et possède ses propres besoins et fonctionnalités qui contribuent à vous and/or aider à gérer le coût global.
  3. Votre diagramme doit inclure l'utilisation du calcul, de la mise en réseau, du stockage, des politiques de cycle de vie, de la collecte de métriques ainsi que les ElastiCache éléments opérationnels et fonctionnels de votre application
  4. Les exigences de votre charge de travail sont susceptibles d'évoluer au fil du temps et il est essentiel que vous continuiez à maintenir et à documenter votre compréhension des composants sous-jacents ainsi que de vos principaux objectifs fonctionnels afin de rester proactif dans la gestion des coûts de votre charge de travail.
  5. Le soutien de la direction en matière de visibilité, de responsabilité, de priorisation et de ressources est essentiel pour que vous disposiez d'une stratégie de gestion des coûts efficace pour votre ElastiCache entreprise.

COÛT 2 : Comment utilisez-vous les outils de surveillance continue pour vous aider à optimiser les coûts associés à vos ElastiCache ressources ?

Introduction au niveau des questions : vous devez rechercher un juste équilibre entre vos indicateurs de ElastiCache coût et de performance des applications. Amazon CloudWatch fournit une visibilité sur les indicateurs opérationnels clés qui peuvent vous aider à déterminer si vos ElastiCache ressources sont surutilisées ou sous-utilisées, par rapport à vos besoins. Du point de vue de l'optimisation des coûts, vous devez comprendre à quel moment vous êtes surapprovisionné et être en mesure de développer des mécanismes appropriés pour redimensionner vos ElastiCache

ressources tout en maintenant vos besoins opérationnels, de disponibilité, de résilience et de performance.

Avantage au niveau de la question : dans l'idéal, vous aurez alloué suffisamment de ressources pour répondre aux besoins opérationnels de votre charge de travail et vous n'aurez pas de ressources sous-utilisées pouvant entraîner un état des coûts sous-optimal. Vous devez être en mesure d'identifier et d'éviter d'exploiter ElastiCache des ressources surdimensionnées pendant de longues périodes.

- [Obligatoire] CloudWatch À utiliser pour surveiller vos ElastiCache clusters et analyser le lien entre ces indicateurs et vos tableaux de bord AWS Cost Explorer.
  1. ElastiCache fournit à la fois des métriques au niveau de l'hôte (par exemple, l'utilisation du processeur) et des métriques spécifiques au logiciel du moteur de cache (par exemple, les mises en cache et les erreurs de cache). Ces métriques sont mesurées et publiées pour chaque nœud de cache toutes les 60 secondes.
  2. ElastiCache les indicateurs de performance (CPUUtilization, EngineUtilization, SwapUsage CurrConnections, et évictions) peuvent indiquer que vous devez effectuer une mise à l'échelle up/down (utiliser des types de nœuds de larger/smaller cache) ou des in/out (add more/less partitions). Comprenez les implications financières des décisions de mise à l'échelle en créant une matrice stratégique qui estime les coûts supplémentaires et les délais minimum et maximum nécessaires pour atteindre les seuils de performance de vos applications.

  [Ressources] :

  - Surveillance de l'utilisation à l'aide de CloudWatch métriques
  - Quelles métriques dois-je surveiller ?
  - ElastiCacheTarifs Amazon
- [Obligatoire] Comprenez et documentez votre stratégie de sauvegarde et ses implications financières.
  1. Avec ElastiCache, les sauvegardes sont stockées dans Amazon S3, qui fournit un stockage durable. Vous devez comprendre les implications financières liées à votre capacité à récupérer en cas de panne.
  2. Activez les sauvegardes automatiques qui supprimeront les fichiers de sauvegarde qui dépassent la limite de conservation.

  [Ressources] :

  - Planification des sauvegardes automatiques

- Tarification Amazon S3

- [Meilleure pratique] Utilisez des nœuds réservés pour vos instances dans le cadre d'une stratégie délibérée visant à gérer les coûts liés à des charges de travail bien comprises et documentées. Des frais initiaux sont facturés pour les nœuds réservés en fonction du type de nœud et de la durée de réservation : un ou trois ans. Ces frais sont largement inférieurs aux frais d'utilisation horaire facturés pour les nœuds à la demande.

  1. Vous devrez peut-être exploiter vos ElastiCache clusters à l'aide de nœuds à la demande jusqu'à ce que vous ayez collecté suffisamment de données pour estimer les exigences en matière d'instances réservées. Planifiez et documentez les ressources nécessaires pour répondre à vos besoins et comparez les coûts attendus selon les types d'instances (à la demande ou réservées)
  2. Évaluez régulièrement les nouveaux types de nœuds de cache disponibles et déterminez s'il est judicieux, du point de vue des métriques de coût et opérationnelles, de migrer votre flotte d'instances vers de nouveaux types de nœuds de cache.

POC 3 : Devez-vous utiliser un type d'instance prenant en charge la hiérarchisation des données ? Quels sont les avantages de la hiérarchisation des données ? Quand ne pas utiliser les instances de hiérarchisation des données ?

Introduction au niveau de la question : la sélection du type d'instance approprié peut non seulement avoir un impact sur les performances et le niveau de service, mais également un impact financier. Les types d'instances sont associés à des coûts différents. La sélection d'un ou de plusieurs types d'instances de grande taille capables de répondre à tous les besoins de stockage en mémoire peut s'avérer une décision naturelle. Cela pourrait toutefois avoir un impact financier important à mesure que le projet arrivera à maturité. Pour s'assurer que le type d'instance sélectionné est correct, il faut examiner périodiquement le temps d'inactivité de ElastiCache l'objet.

Avantage au niveau de la question : vous devez bien comprendre l'impact des différents types d'instances sur vos coûts actuels et futurs. Les modifications marginales ou périodiques de la charge de travail ne doivent pas entraîner de modifications disproportionnées des coûts. Si la charge de travail le permet, les types d'instances qui prennent en charge la hiérarchisation des données offrent un meilleur prix par stockage disponible. En raison du stockage SSD disponible par instance, les instances de hiérarchisation des données prennent en charge une capacité totale de données par instance bien supérieure.

- [Obligatoire] Comprenez les limites des instances de hiérarchisation des données :

1. Disponible uniquement ElastiCache pour les clusters Valkey ou Redis OSS.
2. Seuls certains types d'instances prennent en charge la hiérarchisation des données.
3. Seule ElastiCache la version 6.2 pour Redis OSS et versions ultérieures est prise en charge
4. Les éléments volumineux ne sont pas échangés sur le SSD. Les objets supérieurs à 128 Mio sont conservés en mémoire.

[Ressources] :

- Mise à niveau des données
- ElastiCacheTarifs Amazon

- [Obligatoire] Déterminez quel pourcentage de votre base de données est régulièrement consulté par votre charge de travail.
  1. Les instances de hiérarchisation des données sont idéales pour les charges de travail qui accèdent souvent à une petite partie de votre jeu de données global, mais qui nécessitent tout de même un accès rapide aux données restantes. En d'autres termes, le rapport entre les données chaudes et chaudes est d'environ 20:80.
  2. Développez le suivi du temps d'inactivité des objets au niveau du cluster.
  3. Il est judicieux d'utiliser des implémentations à grande échelle de plus de 500 Go de données.
- [Obligatoire]Sachez que les instances de hiérarchisation des données ne sont pas facultatives pour certaines charges de travail.
  1. L'accès à des objets moins fréquemment utilisés entraîne un faible coût en termes de performances, car ceux-ci sont échangés sur un SSD local. Si votre application est sensible au temps de réponse, testez l'impact sur votre charge de travail.
  2. Elles ne conviennent pas aux caches qui stockent principalement des objets volumineux d'une taille supérieure à 128 Mio.

  [Ressources] :

  - Limites
- [Meilleure pratique] Les types d'instances réservées prennent en charge la hiérarchisation des données. Cela garantit le coût le plus bas en termes de quantité de stockage de données par instance.
  1. Vous devrez peut-être exploiter vos ElastiCache clusters à l'aide d'instances autres que la hiérarchisation des données jusqu'à ce que vous ayez une meilleure compréhension de vos besoins.
  2. Analysez le modèle d'utilisation des données de vos ElastiCache clusters.

3. Créez une tâche automatisée qui collecte régulièrement le temps d'inactivité des objets.
4. Si vous remarquez qu'un pourcentage élevé (environ 80 %) d'objets sont inactifs pendant une période jugée appropriée à votre charge de travail, documentez les résultats et suggérez de migrer le cluster vers des instances prenant en charge la hiérarchisation des données.
5. Évaluez régulièrement les nouveaux types de nœuds de cache disponibles et déterminez s'il est judicieux, du point de vue des métriques de coût et opérationnelles, de migrer votre flotte d'instances vers de nouveaux types de nœuds de cache.

[Ressources] :

- OBJECT IDLETIME
- ElastiCacheTarifs Amazon

Étapes de dépannage courantes et meilleures pratiques avec ElastiCache

Les rubriques suivantes fournissent des conseils de dépannage en cas d'erreurs et de problèmes que vous pourriez rencontrer lors de l'utilisation ElastiCache. Si vous trouvez un problème qui n'est pas répertorié ici, vous pouvez utiliser le bouton de commentaires de cette page pour le signaler.

Pour obtenir des conseils de dépannage supplémentaires et des réponses aux questions d'assistance les plus courantes, consultez le centre de AWS connaissances

Rubriques

- Problèmes de connexion
- Erreurs du client Valkey ou Redis OSS
- Résolution des problèmes de latence élevée en mode ElastiCache Serverless
- Résolution des problèmes de régulation dans Serverless ElastiCache
- Problèmes de connexion persistants
- Rubriques connexes

Problèmes de connexion

Si vous ne parvenez pas à vous connecter à votre ElastiCache cache, envisagez l'une des solutions suivantes :

1. Utilisation du protocole TLS : si votre connexion est bloquée lorsque vous essayez de vous connecter à votre ElastiCache point de terminaison, il se peut que vous n'utilisiez pas le protocole TLS dans votre client. Si vous utilisez ElastiCache Serverless, le chiffrement en transit est toujours activé. Assurez-vous que votre client utilise le protocole TLS pour se connecter au cache. En savoir plus sur la connexion à un cache compatible TLS.
2. VPC : les ElastiCache caches ne sont accessibles que depuis un VPC. Assurez-vous que l' EC2 instance à partir de laquelle vous accédez au cache et le ElastiCache cache sont créés dans le même VPC. Vous devez également activer le peering VPC entre le VPC où EC2 réside votre instance et le VPC dans lequel vous créez votre cache.
3. Groupes de sécurité : ElastiCache utilise des groupes de sécurité pour contrôler l'accès à votre cache. Éléments à prendre en compte :
   a. Assurez-vous que le groupe de sécurité utilisé par votre ElastiCache cache autorise l'accès entrant à celui-ci depuis votre EC2 instance. Cliquez ici pour savoir comment configurer correctement les règles de trafic entrant dans votre groupe de sécurité.
   b. Assurez-vous que le groupe de sécurité utilisé par votre ElastiCache cache autorise l'accès aux ports de votre cache (6379 et 6380 pour le mode sans serveur, et 6379 par défaut pour les ports conçus par vous-même). ElastiCache utilise ces ports pour accepter les commandes Valkey ou Redis OSS. Pour en savoir plus sur la configuration de l'accès aux ports, cliquez ici.

Si la connexion continue d'être difficile, consultez Problèmes de connexion persistants les autres étapes.

Erreurs du client Valkey ou Redis OSS

ElastiCache Le mode Serverless n'est accessible qu'à l'aide de clients prenant en charge le protocole en mode cluster Valkey ou Redis OSS. Les clusters conçus par nos soins sont accessibles depuis les clients dans l'un ou l'autre mode, en fonction de la configuration du cluster.

Si vous rencontrez des erreurs dans votre client, prenez en compte les points suivants :

1. Mode cluster : si vous rencontrez des erreurs CROSSLOT ou des erreurs avec la commande SELECT, vous essayez peut-être d'accéder à un cache activé en mode cluster avec un client Valkey ou Redis OSS qui ne prend pas en charge le protocole Cluster. ElastiCache Serverless ne prend en charge que les clients qui prennent en charge le protocole de cluster Valkey ou Redis OSS. Si vous souhaitez utiliser Valkey ou Redis OSS en mode « Cluster Disabled » (CMD), vous devez concevoir votre propre cluster.

2. Erreurs CROSSLOT : si vous rencontrez cette `ERR CROSSLOT Keys in request don't hash to the same slot` erreur, vous tentez peut-être d'accéder à des clés qui n'appartiennent pas au même emplacement dans un cache en mode cluster. Pour rappel, ElastiCache Serverless fonctionne toujours en mode cluster. Les opérations multiclés, les transactions ou les scripts Lua impliquant plusieurs clés ne sont autorisés que si toutes les clés impliquées se trouvent dans le même emplacement de hachage.

Pour connaître les meilleures pratiques supplémentaires concernant la configuration des clients Valkey ou Redis OSS, veuillez consulter ce billet de blog.

Résolution des problèmes de latence élevée en mode ElastiCache Serverless

Si votre charge de travail semble présenter une latence élevée, vous pouvez analyser les `SuccessfulWriteRequestLatency` métriques CloudWatch `SuccessfulReadRequestLatency` et pour vérifier si la latence est liée au mode ElastiCache Serverless. Ces mesures mesurent la latence interne à ElastiCache Serverless. La latence côté client et les temps de trajet réseau entre votre client et le point de terminaison ElastiCache sans serveur ne sont pas inclus.

Résolution des problèmes de latence côté client

Si vous remarquez une latence élevée côté client, mais aucune augmentation correspondante, CloudWatch `SuccessfulReadRequestLatency` et si `SuccessfulWriteRequestLatency` des mesures mesurent la latence côté serveur, tenez compte des points suivants :

- Assurez-vous que le groupe de sécurité autorise l'accès aux ports 6379 et 6380 : ElastiCache Serverless utilise le port 6379 pour le point de terminaison principal et le port 6380 pour le point de terminaison du lecteur. Certains clients établissent une connectivité aux deux ports pour chaque nouvelle connexion, même si votre application n'utilise pas la fonctionnalité Read from Replica. Si votre groupe de sécurité n'autorise pas l'accès entrant aux deux ports, l'établissement de la connexion peut prendre plus de temps. Pour en savoir plus sur la configuration de l'accès aux ports, cliquez ici.

Résolution des problèmes de latence côté serveur

Une certaine variabilité et des pics occasionnels ne devraient pas être une source de préoccupation. Toutefois, si les `Average` statistiques indiquent une forte augmentation et persistent, vous devriez

consulter le Personal Health Dashboard AWS Health Dashboard et votre Personal Health Dashboard pour plus d'informations. Si nécessaire, pensez à ouvrir un étui de support avec Support.

Tenez compte des meilleures pratiques et stratégies suivantes pour réduire le temps de latence :

- Activer la lecture depuis une réplique : si votre application le permet, nous vous recommandons d'activer la fonctionnalité « Lire depuis une réplique » dans votre client Valkey ou Redis OSS afin de dimensionner les lectures et de réduire la latence. Lorsqu'il est activé, ElastiCache Serverless tente d'acheminer vos demandes de lecture vers des nœuds de cache répliqués situés dans la même zone de disponibilité (AZ) que votre client, évitant ainsi la latence du réseau inter-AZ. Notez que l'activation de la fonctionnalité Read from Replica dans votre client signifie que votre application accepte une éventuelle cohérence des données. Votre application peut recevoir des données plus anciennes pendant un certain temps si vous essayez de les lire après avoir écrit sur une clé.
- Assurez-vous que votre application est déployée au même endroit AZs que votre cache : vous pouvez observer une latence plus élevée côté client si votre application n'est pas déployée au même AZs endroit que votre cache. Lorsque vous créez un cache sans serveur, vous pouvez fournir les sous-réseaux à partir desquels votre application va accéder au cache, et ElastiCache Serverless crée des points de terminaison VPC dans ces sous-réseaux. Assurez-vous que votre application est déployée dans le même environnement AZs. Dans le cas contraire, votre application risque de subir un saut cross-AZ lors de l'accès au cache, ce qui augmentera la latence côté client.
- Réutilisation des connexions : les demandes ElastiCache sans serveur sont effectuées via une connexion TCP compatible TLS à l'aide du protocole RESP. L'établissement de la connexion (y compris l'authentification de la connexion, si elle est configurée) prend du temps, de sorte que la latence de la première demande est supérieure à la normale. Les requêtes via une connexion déjà initialisée offrent une ElastiCache faible latence constante. Pour cette raison, vous devez envisager d'utiliser le regroupement de connexions ou de réutiliser les connexions Valkey ou Redis OSS existantes.
- Vitesse de mise à l'échelle : ElastiCache Serverless évolue automatiquement à mesure que votre taux de demandes augmente. Une augmentation soudaine et importante du taux de requêtes, plus rapide que la vitesse à laquelle ElastiCache Serverless évolue, peut entraîner une latence élevée pendant un certain temps. ElastiCache Les applications sans serveur peuvent généralement augmenter rapidement le taux de demandes prises en charge, ce qui prend jusqu'à 10 à 12 minutes pour doubler le taux de demandes.
- Inspectez les commandes de longue durée : certaines commandes Valkey ou Redis OSS, notamment les scripts Lua ou les commandes sur de grandes structures de données, peuvent

s'exécuter pendant une longue période. Pour identifier ces commandes, ElastiCache publie des métriques au niveau des commandes. Avec ElastiCache Serverless, vous pouvez utiliser les `BasedECPUs` métriques.

- Demandes limitées : lorsque les demandes sont limitées en mode ElastiCache Serverless, vous pouvez constater une augmentation de la latence côté client dans votre application. Lorsque les demandes sont limitées dans ElastiCache Serverless, vous devriez constater une augmentation de la `ThrottledRequests` ElastiCache métrique Serverless. Consultez la section ci-dessous pour résoudre les problèmes liés aux demandes limitées.
- Distribution uniforme des clés et des requêtes : dans ElastiCache Valkey et Redis OSS, une répartition inégale des clés ou des demandes par emplacement peut entraîner un hot slot, ce qui peut entraîner une latence élevée. ElastiCache Serverless prend en charge jusqu'à 30 000 ECPUs/second (90 000 en ECPUs/second cas d'utilisation de Read from Replica) sur un seul emplacement, dans une charge de travail qui exécute des commandes simples. SET/GET Nous vous recommandons d'évaluer la distribution de vos clés et de vos demandes entre les emplacements et de garantir une distribution uniforme si votre taux de demandes dépasse cette limite.

Résolution des problèmes de régulation dans Serverless ElastiCache

Dans les architectures orientées services et les systèmes distribués, la limitation de la vitesse à laquelle les appels d'API sont traités par les différents composants du service est appelée limitation. Cela atténue les pics, contrôle les incohérences dans le débit des composants et permet des restaurations plus prévisibles en cas d'événement opérationnel inattendu. ElastiCache Le mode Serverless est conçu pour ces types d'architectures, et la plupart des clients Valkey ou Redis OSS proposent des tentatives intégrées pour les demandes limitées. Un certain degré de limitation ne constitue pas nécessairement un problème pour votre application, mais la limitation persistante d'une partie sensible à la latence de votre flux de données peut avoir un impact négatif sur l'expérience utilisateur et réduire l'efficacité globale du système.

Lorsque les demandes sont limitées dans ElastiCache Serverless, vous devriez constater une augmentation de la `ThrottledRequests` ElastiCache métrique Serverless. Si vous constatez un nombre élevé de demandes limitées, tenez compte des points suivants :

- Vitesse de mise à l'échelle : le mode ElastiCache sans serveur évolue automatiquement à mesure que vous ingérez de nouvelles données ou que vous augmentez le taux de demandes. Si votre application évolue plus rapidement que la vitesse à laquelle le mode ElastiCache sans serveur évolue, vos demandes peuvent être limitées tandis ElastiCache que le mode sans serveur évolue

pour s'adapter à votre charge de travail. ElastiCache Le mode Serverless permet généralement d'augmenter rapidement la taille de stockage, jusqu'à 10 à 12 minutes pour doubler la taille de stockage de votre cache.

- Distribution uniforme des clés et des requêtes : dans ElastiCache Valkey et Redis OSS, une répartition inégale des clés ou des demandes par emplacement peut entraîner un hot slot. Un hot slot peut entraîner une limitation du nombre de demandes si le taux de demandes pour un seul emplacement dépasse 30 000 ECPUs/second et s'il s'agit d'une charge de travail qui exécute des commandes simples. SET/GET De même, ElastiCache pour Memcached, un raccourci clavier peut entraîner une limitation du nombre de demandes si le taux de demandes dépasse 30 000 €/seconde. ECPUs
- Lire depuis une réplique : si votre application le permet, pensez à utiliser la fonction « Lire depuis une réplique ». La plupart des clients Valkey ou Redis OSS peuvent être configurés pour « dimensionner les lectures » afin de diriger les lectures vers des nœuds de réplication. Cette fonctionnalité vous permet de dimensionner le trafic de lecture. En outre, ElastiCache Serverless achemine automatiquement la lecture depuis les demandes de réplication vers les nœuds situés dans la même zone de disponibilité que votre application, ce qui permet de réduire le temps de latence. Lorsque Read from Replica est activé, vous pouvez atteindre jusqu'à 90 000 points ECPUs/second sur un seul emplacement, pour des charges de travail utilisant de simples SET/GET commandes.

Problèmes de connexion persistants

Les éléments suivants doivent être vérifiés lors de la résolution des problèmes de connectivité persistants avec ElastiCache :

Rubriques

- Groupes de sécurité
- Réseau ACLs
- Tables de routage
- Résolution DNS
- Identification des problèmes liés aux diagnostics côté serveur
- Validation de connectivité réseau
- Limites liées au réseau
- Utilisation de l'UC

- Connexions résiliées côté serveur
- Résolution des problèmes côté client pour les instances Amazon EC2
- Disséquer le temps nécessaire pour compléter une seule demande

Groupes de sécurité

Les groupes de sécurité sont des pare-feux virtuels qui protègent votre ElastiCache client (EC2 instance, AWS Lambda fonction, conteneur Amazon ECS, etc.) et votre ElastiCache cache. Les groupes de sécurité sont avec état, ce qui signifie qu'une fois le trafic entrant ou sortant autorisé, les réponses pour ce trafic seront automatiquement autorisées dans le contexte de ce groupe de sécurité spécifique.

La fonction avec état exige que le groupe de sécurité suive constamment toutes les connexions autorisées, et il existe une limite pour les connexions suivies. Si la limite est atteinte, les nouvelles connexions échoueront. Reportez-vous à la section de résolution des problèmes pour savoir comment déterminer si les limites ont été atteintes du côté du client ou du ElastiCache côté du client.

Vous pouvez avoir un seul groupe de sécurité attribué en même temps au client et au ElastiCache cluster, ou des groupes de sécurité individuels pour chacun d'entre eux.

Dans les deux cas, vous devez autoriser le trafic sortant TCP sur le ElastiCache port depuis la source et le trafic entrant sur le même port à destination. ElastiCache Le port par défaut est 11211 pour Memcached et 6379 pour Valkey ou Redis OSS. Par défaut, les groupes de sécurité autorisent la totalité du trafic sortant. Dans ce cas, seule la règle entrante dans le groupe de sécurité cible est requise.

Pour plus d'informations, consultez Modèles d'accès pour accéder à un ElastiCache cluster dans un Amazon VPC.

Réseau ACLs

Les listes de contrôle d'accès réseau (ACLs) sont des règles apatrides. Le trafic doit être autorisé dans les deux sens (entrant et sortant) pour réussir. ACLs Les réseaux sont affectés à des sous-réseaux, et non à des ressources spécifiques. Il est possible que la même ACL soit assignée ElastiCache et que la même ressource client soit affectée, surtout s'ils se trouvent dans le même sous-réseau.

Par défaut, le réseau ACLs autorise tout le trafic. Toutefois, il est possible de les personnaliser pour refuser ou autoriser le trafic. En outre, l'évaluation des règles ACL est séquentielle, ce qui

signifie que la règle avec le nombre le plus bas correspondant au trafic l'autorisera ou le refusera. La configuration minimale pour autoriser le trafic Valkey ou Redis OSS est la suivante :

ACL réseau côté client :

- Règles entrantes :
- Numéro de règle : de préférence inférieur à toute règle de refus ;
- Type : Règle TCP personnalisée ;
- Protocole : TCP
- Plage de ports : 1024-65535
- Source : 0.0.0.0/0 (ou créez des règles individuelles pour les sous-réseaux du cluster) ElastiCache
- Autoriser/Refuser : Autoriser

- Règles sortantes :
- Numéro de règle : de préférence inférieur à toute règle de refus ;
- Type : Règle TCP personnalisée ;
- Protocole : TCP
- Plage de ports : 6379
- Source : 0.0.0.0/0 (ou les sous-réseaux du ElastiCache cluster. (N'oubliez pas que l'utilisation de produits spécifiques IPs peut créer des problèmes en cas de basculement ou de dimensionnement du cluster)
- Autoriser/Refuser : Autoriser

ElastiCache ACL réseau :

- Règles entrantes :
- Numéro de règle : de préférence inférieur à toute règle de refus ;
- Type : Règle TCP personnalisée ;
- Protocole : TCP
- Plage de ports : 6379
- Source : 0.0.0.0/0 (ou créez des règles individuelles pour les sous-réseaux du cluster) ElastiCache
- Autoriser/Refuser : Autoriser

- Règles sortantes :
- Numéro de règle : de préférence inférieur à toute règle de refus ;
- Type : Règle TCP personnalisée ;
- Protocole : TCP
- Plage de ports : 1024-65535
- Source : 0.0.0.0/0 (ou les sous-réseaux du ElastiCache cluster. (N'oubliez pas que l'utilisation de produits spécifiques IPs peut créer des problèmes en cas de basculement ou de dimensionnement du cluster)
- Autoriser/Refuser : Autoriser

Pour plus d'informations, consultez la section Réseau ACLs.

Tables de routage

Comme dans le cas du réseau ACLs, chaque sous-réseau peut avoir des tables de routage différentes. Si les clients et le ElastiCache cluster se trouvent dans des sous-réseaux différents, assurez-vous que leurs tables de routage leur permettent de communiquer entre eux.

Les environnements plus complexes, impliquant plusieurs VPCs routeurs dynamiques ou pare-feux réseau, peuvent s'avérer difficiles à dépanner. Veuillez consulter Validation de connectivité réseau pour confirmer que vos paramètres réseau sont appropriés.

Résolution DNS

ElastiCache fournit les points de terminaison du service en fonction des noms DNS. Les points de terminaison disponibles sont `Configuration`, `Primary`, `Reader` et `Node`. Pour plus d'informations, voir Recherche de points de terminaison de connexion.

En cas de basculement ou de modification de cluster, l'adresse associée au nom du point de terminaison peut changer et sera automatiquement mise à jour.

Les paramètres DNS personnalisés (c'est-à-dire ceux qui n'utilisent pas le service DNS VPC) ne connaissent peut-être pas les noms DNS ElastiCache fournis. Assurez-vous que votre système parvient à résoudre les ElastiCache points de terminaison à l'aide d'outils système tels que `dig` (comme indiqué ci-dessous) ou`nslookup`.

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
```

```
1.2.3.4
```

Vous pouvez également forcer la résolution de noms via le service DNS VPC :

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Identification des problèmes liés aux diagnostics côté serveur

CloudWatch les métriques et les informations d'exécution du ElastiCache moteur sont des sources ou informations courantes permettant d'identifier les sources potentielles de problèmes de connexion. Une bonne analyse commence généralement par les éléments suivants :

- Utilisation du processeur : Valkey et Redis OSS sont des applications multithread. Cependant, l'exécution de chaque commande se produit dans un seul thread (principal). Pour cette raison, ElastiCache fournit les métriques `CPUUtilization` et`EngineCPUUtilization`. `EngineCPUUtilization`fournit l'utilisation du processeur dédiée au processus Valkey ou Redis OSS, et `CPUUtilization` l'utilisation dans tous les v. CPUs Les nœuds dotés de plusieurs vCPU ont généralement des valeurs différentes pour `CPUUtilization` et`EngineCPUUtilization`, la seconde étant généralement plus élevée. Une `EngineCPUUtilization` élevée peut être causée par un nombre élevé de requêtes ou d'opérations complexes qui prennent beaucoup de temps d'UC à finaliser. Vous pouvez identifier les deux avec les éléments suivants :
  - Nombre élevé de demandes : recherchez des augmentations sur d'autres métriques correspondant au modèle `EngineCPUUtilization`. Les métriques utiles sont les suivantes :
    - `CacheHits` et `CacheMisses` : le nombre de demandes réussies ou de demandes qui n'ont pas trouvé d'élément valide dans le cache. Si le ratio des manquements par rapport aux résultats est élevé, l'application perd du temps et des ressources avec des demandes infructueuses.
    - `SetTypeCmds` et `GetTypeCmds` : ces métriques en corrélation avec `EngineCPUUtilization` peuvent aider à comprendre si la charge est significativement plus élevée pour les demandes d'écriture, mesurées par `SetTypeCmds`, ou les lectures, mesurées par `GetTypeCmds`. Si la charge est principalement composée de lectures, l'utilisation de plusieurs réplicas en lecture peut équilibrer les demandes sur plusieurs nœuds et épargner le principal pour les écritures. Dans les clusters désactivés en mode cluster, l'utilisation de répliques en lecture peut être effectuée en créant une configuration de connexion supplémentaire dans l'application à l'aide du point de terminaison du lecteur. ElastiCache Pour

plus d'informations, voir Recherche de points de terminaison de connexion. Les opérations de lecture doivent être soumises à cette connexion supplémentaire. Les opérations d'écriture seront effectuées via le point de terminaison principal normal. En mode cluster activé, il est conseillé d'utiliser une bibliothèque qui prend en charge les réplicas en lecture de façon native. Avec les bons indicateurs, la bibliothèque sera en mesure de découvrir automatiquement la topologie du cluster, les nœuds de réplication, d'activer les opérations de lecture via la commande READONLY Valkey ou Redis OSS et de soumettre les demandes de lecture aux répliques.

- Nombre élevé de connexions :
  - `CurrConnections` et `NewConnections` : `CurrConnection` est le nombre de connexions établies au moment de la collectie de données, tandis que`NewConnections` indique le nombre de connexions créées au cours de la période.

    La création et la gestion des connexions impliquent une surcharge d'UC importante. En outre, la connexion en trois temps TCP requise pour créer des connexions affectera négativement les temps de réponse globaux.

    Un ElastiCache nœud avec des milliers `NewConnections` par minute indique qu'une connexion est créée et utilisée par quelques commandes seulement, ce qui n'est pas optimal. Le fait de maintenir les connexions établies et de les réutiliser pour de nouvelles opérations est une bonne pratique. Ceci est possible lorsque l'application client prend en charge et implémente correctement le regroupement de connexions ou les connexions persistantes. Avec le regroupement de connexions, le nombre de `currConnections` n'a pas de grandes variations et les `NewConnections` doivent être aussi faibles que possible. Valkey et Redis OSS offrent des performances optimales avec un petit nombre de CurrConnections. En maintenant currConnection dans l'ordre de dizaines ou de centaines, on minimise l'utilisation des ressources nécessaires à la prise en charge des connexions individuelles, comme les tampons client et les cycles d'UC pour servir la connexion.
- Débit réseau :
  - Déterminez la bande passante : la bande passante du réseau des ElastiCache nœuds est proportionnelle à la taille du nœud. Comme les applications ont des caractéristiques différentes, les résultats peuvent varier en fonction de la charge de travail. Par exemple, les applications avec un nombre élevé de petites demandes ont tendance à affecter davantage l'utilisation de l'UC que le débit réseau, tandis que les clés plus grandes provoqueront une utilisation plus élevée du réseau. Pour cette raison, il est conseillé de tester les nœuds avec la charge de travail réelle pour une meilleure compréhension des limites.

La simulation de la charge à partir de l'application donnerait des résultats plus précis. Cependant, les outils de référence peuvent donner une bonne idée des limites.

- Dans les cas où les requêtes sont principalement des lectures, l'utilisation de réplicas pour les opérations de lecture allégera la charge sur le nœud primaire. Si le cas d'utilisation est principalement des écritures, l'utilisation de nombreux réplicas amplifiera l'utilisation du réseau. Pour chaque octet écrit sur le nœud primaire, N octets seront envoyés aux réplicas, N étant le nombre de réplicas. La meilleure pratique pour les charges de travail intensives en écriture consiste à utiliser ElastiCache Redis OSS avec le mode cluster activé afin que les écritures puissent être équilibrées sur plusieurs partitions ou être étendues à un type de nœud offrant davantage de capacités réseau.
- Les CloudWatchmetrics `NetworkBytesIn` et `NetworkBytesOut` indiquent la quantité de données entrant ou sortant du nœud, respectivement. `ReplicationBytes`est le trafic dédié à la réplication des données.

Pour de plus amples informations, veuillez consulter Limites liées au réseau.

- Commandes complexes : les commandes Redis OSS sont diffusées sur un seul thread, ce qui signifie que les demandes sont traitées de manière séquentielle. Une seule commande lente peut affecter d'autres demandes et connexions, aboutissant ainsi à des délais d'expiration. L'utilisation de commandes qui agissent sur plusieurs valeurs, clés ou types de données doit être effectuée avec soin. Les connexions peuvent être bloquées ou interrompues en fonction du nombre de paramètres ou de la taille de ses valeurs d'entrée ou de sortie.

  Un exemple bien connu est la commande `KEYS`. Il balaie le keyspace entier en recherchant un modèle donné et bloque l'exécution d'autres commandes lors de son exécution. Redis OSS utilise la notation « Big O » pour décrire la complexité de ses commandes.

  La commande de clés a une complexité temporelle O(N), N étant le nombre de clés dans la base de données. Par conséquent, plus le nombre de clés est grand, plus la commande sera lente. `KEYS` peut causer des problèmes de différentes manières : si aucun modèle de recherche n'est utilisé, la commande retournera tous les noms de clés disponibles. Dans les bases de données contenant des milliers ou des millions d'éléments, une sortie énorme sera créée et inondera les tampons réseau.

  Si un modèle de recherche est utilisé, seules les clés correspondant au modèle seront retournées au client. Cependant, le moteur balaie toujours le keyspace entier pour le rechercher, et le temps nécessaire pour finaliser la commande sera le même.

Une alternative pour KEYS est la commande SCAN. Elle itère sur le keyspace et limite les itérations dans un nombre spécifique d'éléments, en évitant les blocs prolongés sur le moteur.

L'analyse dispose du paramètre COUNT, utilisé pour définir la taille des blocs d'itération. La valeur par défaut est de 10 (10 éléments par itération).

Selon le nombre d'éléments dans la base de données, les petits blocs de valeur COUNT nécessitent plus d'itérations pour effectuer une analyse complète, tandis que des valeurs plus grandes garderont le moteur occupé plus longtemps à chaque itération. Alors que les petites valeurs de nombre ralentiront SCAN sur les grandes bases de données, des valeurs plus grandes peuvent causer les mêmes problèmes que ceux mentionnés pour KEYS.

À titre d'exemple, l'exécution de la commande SCAN avec la valeur de nombre 10 nécessite 100 000 répétitions sur une base de données avec 1 million de clés. Si le temps de transmission aller-retour moyen du réseau est de 0,5 milliseconde, environ 50 000 millisecondes (50 secondes) seront consacrées au transfert de demandes.

D'autre part, si la valeur de comptage était de 100 000, une seule itération serait nécessaire et seulement 0,5 ms serait dépensé pour le transférer. Cependant, le moteur serait complètement bloqué pour d'autres opérations jusqu'à ce que la commande ait fini de balayer tout le keyspace.

En outre KEYS, plusieurs autres commandes sont potentiellement nuisibles si elles ne sont pas utilisées correctement. Pour voir la liste de toutes les commandes et leur complexité temporelle respective, accédez aux commandes Valkey et Redis OSS.

Exemples de problèmes potentiels :

- Scripts Lua : Valkey et Redis OSS fournissent un interpréteur Lua intégré, permettant l'exécution de scripts côté serveur. Les scripts Lua sur Valkey et Redis OSS sont exécutés au niveau du moteur et sont atomiques par définition, ce qui signifie qu'aucune autre commande ou script ne sera autorisé à s'exécuter pendant qu'un script est en cours d'exécution. Les scripts Lua offrent la possibilité d'exécuter plusieurs commandes, algorithmes de prise de décision, analyse de données, etc. directement sur le moteur. Alors que l'atomicité des scripts et les chances de décharger l'application sont tentantes, les scripts doivent être utilisés avec prudence et pour de petites opérations. ElastiCacheActivé, le temps d'exécution des scripts Lua est limité à 5 secondes. Les scripts qui n'ont pas été écrits dans le keyspace seront automatiquement résiliés après la période de 5 secondes. Pour éviter la corruption des données et les incohérences, le nœud basculera si l'exécution du script n'a pas été terminée en 5 secondes et a eu une écriture pendant son exécution. Les transactions sont l'alternative

pour garantir la cohérence des multiples modifications clés associées dans Redis OSS. Une transaction permet l'exécution d'un bloc de commandes, en surveillant les clés existantes pour les modifications. Si l'une des clés surveillées change avant la fin de la transaction, toutes les modifications sont supprimées.

- Suppression en masse d'éléments : la commande `DEL` accepte plusieurs paramètres, qui sont les noms de clés à supprimer. Les opérations de suppression sont synchrones et prennent beaucoup de temps d'UC si la liste des paramètres est volumineuse ou contient une liste, un ensemble, un ensemble trié ou un hachage de grande taille (structures de données contenant plusieurs sous-éléments). En d'autres termes, même la suppression d'une seule clé peut prendre beaucoup de temps si elle comporte de nombreux éléments. L'alternative à `DEL` is est `UNLINK` une commande asynchrone disponible depuis Redis OSS 4. `UNLINK`doit être préféré dans la `DEL` mesure du possible. À partir ElastiCache de Redis OSS 6x, le `lazyfree-lazy-user-del` paramètre fait en sorte que la `DEL` commande se comporte comme si `UNLINK` elle était activée. Pour plus d'informations, consultez la section Modifications des paramètres de Redis OSS 6.0.
- Commandes agissant sur plusieurs clés : `DEL` a été mentionné précédemment comme une commande qui accepte plusieurs arguments et son temps d'exécution sera directement proportionnel à cela. Cependant, Redis OSS fournit de nombreuses autres commandes qui fonctionnent de la même manière. À titre d'exemple, `MSET` et `MGET` permettent l'insertion ou la récupération de plusieurs clés de chaîne à la fois. Leur utilisation peut être bénéfique pour réduire la latence réseau inhérente à plusieurs commandes `SET` ou `GET`. Cependant, une liste exhaustive de paramètres affectera l'utilisation de l'UC.

  Bien que l'utilisation de l'UC seule ne soit pas la cause des problèmes de connectivité, passer trop de temps pour traiter une seule ou quelques commandes sur plusieurs clés peut provoquer l'échec d'autres demandes et augmenter l'utilisation globale de l'UC.

  Le nombre de clés et leur taille affecteront la complexité de la commande et, par conséquent, le temps d'exécution.

  Autres exemples de commandes qui peuvent agir sur plusieurs clés : `HMGET`, `HMSET`, `MSETNX`, `PFCOUNT`, `PFMERGE`, `SDIFF`, `SDIFFSTORE`, `SINTER`, `SINTERSTORE`, `SUNION`, `SUNIONSTORE`, `TOUCH`, `ZDIFF`, `ZDIFFSTORE`, `ZINTER` ou `ZINTERSTORE`.
- Commandes agissant sur plusieurs types de données : Redis OSS fournit également des commandes qui agissent sur une ou plusieurs touches, quel que soit leur type de données. ElastiCache pour Redis OSS fournit la métrique permettant `KeyBasedCmds` de surveiller ces

commandes. Cette métrique additionne l'exécution des commandes suivantes dans la période sélectionnée :

- Complexité O(N) :
  - `KEYS`
- O(1)
  - `EXISTS`
  - `OBJECT`
  - `PTTL`
  - `RANDOMKEY`
  - `TTL`
  - `TYPE`
  - `EXPIRE`
  - `EXPIREAT`
  - `MOVE`
  - `PERSIST`
  - `PEXPIRE`
  - `PEXPIREAT`
  - `UNLINK (O(N)` pour récupérer de la mémoire. Cependant, la tâche de récupération de mémoire se produit dans un thread séparé et ne bloque pas le moteur
- Différents temps de complexité en fonction du type de données :
  - `DEL`
  - `DUMP`
  - `RENAME` est considéré comme une commande avec une complexité O(1), mais exécute `DEL` en interne. Le temps d'exécution varie en fonction de la taille de la clé renommée.
  - `RENAMENX`
  - `RESTORE`
  - `SORT`
- Gros hachages : le hachage est un type de données qui autorise une seule clé avec plusieurs sous-éléments clé-valeur. Chaque hachage peut stocker 4 294 967 295 éléments et les opérations sur de gros hachages peuvent devenir coûteuses. De même que pour `KEYS`, les hachages ont la commande `HKEYS` avec une complexité temporelle O(N), N étant

le nombre d'éléments dans le hachage. `HSCAN` doit avoir la préférence sur `HKEYS` pour éviter les commandes à exécution longue. `HDEL`, `HGETALL`, `HMGET`, `HMSET` et `HVALS` sont des commandes qui doivent être utilisées avec prudence sur les gros hachages.

- Autres structures big data : outre les hachages, d'autres structures de données peuvent être intensives en UC. Les ensembles, les listes, les ensembles triés et les Hyperloglogs peuvent également prendre beaucoup de temps à être manipulés en fonction de leur taille et des commandes utilisées. Pour plus d'informations sur ces commandes, consultez les commandes Valkey et Redis OSS.

Validation de connectivité réseau

Après avoir examiné les configurations réseau liées à la résolution DNS, aux groupes de sécurité, au réseau et aux tables de routage ACLs, la connectivité peut être validée à l'aide de l'Analyzer de Reachability VPC et des outils système.

Reachability Analyzer testera la connectivité réseau et confirmera que toutes les exigences et autorisations sont satisfaites. Pour les tests ci-dessous, vous aurez besoin de l'ENI ID (Elastic Network Interface Identification) de l'un des ElastiCache nœuds disponibles dans votre VPC. Vous pouvez le trouver en procédant comme suit :

1. Passer à la https://console.aws.amazon.com/ec2/version 2/home ? #NIC :
2. Filtrez la liste des interfaces en fonction du nom de votre ElastiCache cluster ou de l'adresse IP obtenue lors des validations DNS précédentes.
3. Notez ou enregistrez l'ID ENI. Si plusieurs interfaces sont affichées, consultez la description pour vérifier qu'elles appartiennent au bon ElastiCache cluster et choisissez-en une.
4. Passez à l'étape suivante.
5. Créer un parcours d'analyse https://console.aws.amazon.com/vpc/chez vous ? # ReachabilityAnalyzer et choisissez les options suivantes :

   - Type de source : Choisissez l'instance si votre ElastiCache client fonctionne sur une EC2 instance Amazon ou une interface réseau (s'il utilise un autre service, tel qu' AWS Fargate Amazon ECS avec le réseau awsvpc AWS Lambda, etc.), et l'ID de ressource correspondant (EC2 instance ou ID ENI) ;
   - Type de destination : choisissez Network Interface (Interface réseau) et sélectionnez l'ENI Elasticache dans la liste.

- Port de destination : spécifiez 6379 ElastiCache pour Redis OSS ou 11211 pour ElastiCache Memcached. Ce sont les ports définis avec la configuration par défaut et cet exemple suppose qu'ils ne sont pas modifiés.
- Protocole : TCP

Créez le chemin d'analyse et attendez le résultat quelques instants. Si le statut n'est pas accessible, ouvrez les détails de l'analyse et veuillez consulter l'explorateur d'analyse pour plus de détails sur les cas où les demandes ont été bloquées

Si les tests d'accessibilité aboutissent, passez à la vérification au niveau du système d'exploitation.

Pour valider la connectivité TCP sur le port de ElastiCache service : sur Amazon Linux, elle `Nping` est disponible dans le package `nmap` et permet de tester la connectivité TCP sur le ElastiCache port, tout en fournissant le temps d'aller-retour entre le réseau et l'établissement de la connexion. Utilisez-le pour valider la connectivité réseau et la latence actuelle du ElastiCache cluster, comme indiqué ci-dessous :

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
(Output suppressed )

Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

Par défaut, `nping` envoie 5 sondes avec un délai de 1 seconde entre elles. Vous pouvez utiliser l'option « -c » pour augmenter le nombre de sondes et « --delay » pour modifier la durée avant l'envoi d'un nouveau test.

Si les tests avec `nping` échouent et si les tests VPC Reachability Analyzer aboutissent, demandez à votre administrateur système d'examiner les règles de pare-feu basées sur l'hôte, les règles de routage asymétrique ou toute autre restriction possible au niveau du système d'exploitation.

Sur la ElastiCache console, vérifiez si le chiffrement en transit est activé dans les détails de votre ElastiCache cluster. Si le chiffrement en transit est activé, vérifiez si la session TLS peut être établie à l'aide de la commande suivante :

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Une sortie étendue est attendue si la connexion et la négociation TLS aboutissent. Vérifiez le code de retour disponible sur la dernière ligne, la valeur doit être `0 (ok)`. Si openssl renvoie quelque chose de différent, vérifiez la raison de l'erreur sur https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS.

Si tous les tests d'infrastructure et de système d'exploitation ont réussi mais que votre application ne parvient toujours pas à se connecter ElastiCache, vérifiez si les configurations de l'application sont conformes aux ElastiCache paramètres. Les erreurs les plus courantes sont les suivantes :

- Votre application ne prend pas en charge le mode ElastiCache cluster et le mode cluster ElastiCache est activé ;
- Votre application ne prend pas en charge le protocole TLS/SSL et le chiffrement en transit ElastiCache est activé ;
- L'application prend en charge TLS/SSL mais ne dispose pas des indicateurs de configuration appropriés ni des autorités de certification fiables ;

Limites liées au réseau

- Nombre maximal de connexions : il existe des limites strictes pour les connexions simultanées. Chaque ElastiCache nœud permet jusqu'à 65 000 connexions simultanées entre tous les clients. Cette limite peut être surveillée à l'aide `CurrConnections` des métriques activées CloudWatch. Cependant, les clients ont également leurs limites pour les connexions sortantes. Sous Linux, vérifiez la plage de ports éphémères autorisée à l'aide de la commande :

  ```
  # sysctl net.ipv4.ip_local_port_range
  net.ipv4.ip_local_port_range = 32768 60999
  ```

  Dans l'exemple précédent, 28231 connexions seront autorisées à partir de la même source, vers la même adresse IP de destination (ElastiCache nœud) et le même port. La commande suivante indique le nombre de connexions existantes pour un ElastiCache nœud spécifique (IP 1.2.3.4) :

  ```
  ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
   '^State' | wc -l
  ```

Si le nombre est trop élevé, votre système peut devenir surchargé en essayant de traiter les demandes de connexion. Il est conseillé d'envisager d'implémenter des techniques telles que le regroupement de connexions ou les connexions persistantes pour mieux gérer les connexions. Dans la mesure du possible, configurez le groupe de connexions pour limiter le nombre maximal de connexions à quelques centaines. En outre, une logique d'interruptions exponentielles pour gérer les expirations de délai d'attente ou d'autres exceptions de connexion serait souhaitable pour éviter l'abandon de connexion en cas de problème.

- Limites du trafic réseau : vérifiez les CloudWatch indicateurs suivants pour Redis OSS afin d'identifier les limites réseau possibles sur le ElastiCache nœud :
  - `NetworkBandwidthInAllowanceExceeded` / `NetworkBandwidthOutAllowanceExceeded` : paquets réseau formés parce que le débit dépassait la limite de bande passante agrégée.

    Il est important de noter que chaque octet écrit sur le nœud primaire sera répliqué sur N réplicas, N étant le nombre de réplicas. Les clusters avec de petits types de nœuds, plusieurs réplicas et des demandes d'écriture intensives peuvent ne pas être en mesure de faire face à l'arriéré de réplication. Dans de tels cas, la bonne pratique consiste à effectuer une augmentation d'échelle (changer le type de nœud), une évolutivité horizontale (ajouter des partitions dans des clusters activés en mode cluster), de réduire le nombre de réplicas ou de réduire le nombre d'écritures.
  - `NetworkConntrackAllowanceExceeded` : paquets formés, car le nombre maximal de connexions suivies dans tous les groupes de sécurité affectés au nœud a été dépassé. Les nouvelles connexions échoueront probablement pendant cette période.
  - `NetworkPackets PerSecondAllowanceExceeded` : nombre maximal de paquets par seconde dépassé. Les charges de travail basées sur un taux élevé de très petites demandes peuvent atteindre cette limite avant la bande passante maximale.

Les mesures ci-dessus sont le moyen idéal de confirmer que les nœuds atteignent leurs limites réseau. Cependant, les limites sont également identifiables par des plateaux sur les métriques du réseau.

Si les plateaux sont observés pendant des périodes prolongées, ils seront probablement suivis d'un retard de réplication, d'une augmentation des octets utilisés pour le cache, d'une chute de la mémoire libérable, d'une hausse des échanges et de l'utilisation de l'UC. EC2 Les instances Amazon ont également des limites réseau qui peuvent être suivies grâce aux métriques des pilotes ENA. Les instances Linux avec prise en charge réseaux améliorée et les pilotes ENA

version 2.2.10 ou plus récente peuvent consulter les compteurs de limite avec la commande suivante :

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

Utilisation de l'UC

La métrique d'utilisation du processeur est le point de départ de l'enquête, et les éléments suivants peuvent aider à réduire les éventuels problèmes ElastiCache secondaires :

- Redis OSS SlowLogs : La configuration ElastiCache par défaut conserve les 128 dernières commandes dont l'exécution a pris plus de 10 millisecondes. L'historique des commandes lentes est conservé pendant l'exécution du moteur et sera perdu en cas d'échec ou de redémarrage. Si la liste atteint 128 entrées, les anciens événements seront supprimés pour faire de la place aux nouveaux. La taille de la liste des événements lents et le temps d'exécution considéré comme lent peuvent être modifiés via les paramètres `slowlog-max-len` et `slowlog-log-slower-than` dans un groupe de paramètres personnalisé. La liste des journaux lents peut être récupérée en exécutant `SLOWLOG GET 128` sur le moteur, 128 étant les 128 dernières commandes lentes signalées. Chaque entrée contient les champs suivants :

```
1) 1) (integer) 1 -----------> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -------------> Command
      2) "*" ----------------> Arguments
   5) "1.2.3.4:57004"-> Source
```

L'événement ci-dessus s'est produit le 26 décembre à 19:26:07 UTC, a pris 4,8 secondes (4 823 ms) et a été causé par la commande `KEYS` demandée auprès du client 1.2.3.4.

Sous Linux, l'horodatage peut être converti avec la date de commande :

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Avec Python :

```
>>> from datetime import datetime
```

```
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Ou sous Windows avec PowerShell :

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
DateTime      : 12/26/2020 7:26:07 PM
UtcDateTime
                : 12/26/2020 7:26:07 PM
LocalDateTime : 12/26/2020 2:26:07 PM
Date          : 12/26/2020 12:00:00 AM
Day           : 26
DayOfWeek
                : Saturday
DayOfYear     : 361
Hour          : 19
Millisecond   : 0
Minute        : 26
Month
                : 12
Offset        : 00:00:00Ticks         : 637446075670000000
UtcTicks
                : 637446075670000000
TimeOfDay     : 19:26:07
Year          : 2020
```

Beaucoup de commandes lentes dans un court laps de temps (même minute ou moins) sont une source de préoccupation. Examinez la nature des commandes et la façon dont elles peuvent être optimisées (voir les exemples précédents). Si des commandes avec une complexité temporelle O(1) sont fréquemment signalées, vérifiez les autres facteurs d'utilisation élevée de l'UC mentionnés précédemment.

- Métriques de latence : ElastiCache pour Redis, OSS fournit CloudWatch des métriques pour surveiller la latence moyenne pour différentes classes de commandes. Le point de données est calculé en divisant le nombre total d'exécutions de commandes dans la catégorie par le temps total d'exécution de la période. Il est important de comprendre que les résultats des métriques de latence sont un agrégat de commandes multiples. Une seule commande peut entraîner des résultats inattendus, tels que des délais d'expiration, sans impact significatif sur les métriques. Dans de tels cas, les événements de journaux lents constitueraient une source d'information plus précise. La liste suivante contient les métriques de latence disponibles et les commandes correspondantes qui les affectent.

- EvalBasedCmdsLatency: lié aux commandes Lua Script,`eval`, ; `evalsha`
- GeoSpatialBasedCmdsLatency: `geodist`, `geohash`, `geopos`, `georadius`, `georadiusbymember`, `geoadd`;
- GetTypeCmdsLatency: commandes de lecture, quel que soit le type de données ;
- HashBasedCmdsLatency: `hexists`, `hget`, `hgetall`, `hkeys`, `hlen`, `hmget`, `hvals`, `hstrlen`, `hdel`, `hincrby`, `hincrbyfloat`, `hmset`, `hset`, `hsetnx`;
- HyperLogLogBasedCmdsLatency: `pfselftest`, `pfcount`, `pfdebug`, `pfadd`, `pfmerge`;
- KeyBasedCmdsLatency: commandes qui peuvent agir sur différents types de données :`dump`,`exists`,`keys`,`object`,`pttl`,,`randomkey`,`ttl`,`type`,`del`,`expire`,`expireat`,`move`,`pe` `unlink` ;
- ListBasedCmdsLatency: lindex, len, lrange, blpop, brpoplpush, linsert, lpop, lpush, lpushx, lrem, lset, ltrim, rpop, roplpush, rpush, rpushx ;
- PubSubBasedCmdsLatency: psubscribe, publish, pubsub, punsubscribe, subscribe, unsubscribe ;
- SetBasedCmdsLatency: `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `sunion`, `sadd`, `sdiffstore`, `sinterstore`, `smove`, `spop`, `srem`, `sunionstore`;
- SetTypeCmdsLatency: écrire des commandes, quel que soit le type de données ;
- SortedSetBasedCmdsLatency: `zcard`, `zcount`, `zrange`, `zrangebyscore`, `zrank`, `zrevrange`, `zrevrangebyscore`, `zrevrank`, `zscore`, `zrangebylex`, `zrevrangebylex`, `zlexcount`, `zadd`. `zincrby`, `zinterstore`, `zrem`, `zremrangebyrank`, `zremrangebyscore`, `zunionstore`, `zremrangebylex`, `zpopmax`, `zpopmin`, `bzpopmin`, `bzpopmax`;
- StringBasedCmdsLatency: `bitcount`, `get`, `getbit`, `getrange`, `mget`, `strlen`, `substr`, `bitpos`, `append`, `bitop`, `bitfield`, `decr`, `decrby`, `getset`, `incr`, `incrby`, `incrbyfloat`, `mset`, `msetnx`, `psetex`, `set`, `setbit`, `setex`, `setnx`, `setrange`;
- StreamBasedCmdsLatency: `xrange`, `xrevrange`, `xlen`, `xread`, `xpending`, `xinfo`, `xadd`, `xgroup`, `readgroup`, `xack`, `xclaim`, `xdel`, `xtrim`, `xsetid`;

- Commandes d'exécution de Redis OSS :
  - info commandstats : fournit une liste des commandes exécutées depuis le démarrage du moteur, leur nombre cumulé d'exécutions, leur durée d'exécution totale et le temps d'exécution moyen par commande ;
  - client list : fournit une liste des clients actuellement connectés et des informations pertinentes comme l'utilisation des tampons, la dernière commande exécutée, etc.

- Sauvegarde et réplication : ElastiCache pour les versions de Redis OSS antérieures à 2.8.22, utilisez un processus bifurqué pour créer des sauvegardes et effectuer des synchronisations complètes avec les répliques. Cette méthode peut entraîner une surcharge de mémoire importante pour les cas d'utilisation intensifs en écriture.

  À partir de ElastiCache Redis OSS 2.8.22, une méthode de sauvegarde et de réplication sans fourche a AWS été introduite. La nouvelle méthode peut retarder les écritures afin d'éviter les échecs. Les deux méthodes peuvent provoquer des périodes d'utilisation de l'UC plus élevée, conduire à des temps de réponse plus élevés et, par conséquent, conduire à des délais d'attente du client pendant leur exécution. Vérifiez toujours si les échecs du client se produisent pendant la fenêtre de sauvegarde ou si la métrique `SaveInProgress` était de 1 au cours de la période. Il est conseillé de planifier la fenêtre de sauvegarde pour les périodes de faible utilisation afin de minimiser les risques de problèmes avec les clients ou d'échecs de sauvegarde.

Connexions résiliées côté serveur

La configuration par défaut ElastiCache de Redis OSS maintient les connexions client établies indéfiniment. Toutefois, dans certains cas, la résiliation de la connexion peut être souhaitable. Par exemple :

- Les bogues dans l'application client peuvent entraîner l'oubli de connexions et leur maintien avec un état inactif. C'est ce que l'on appelle une « fuite de connexion » et sa conséquence est une augmentation constante du nombre de connexions établies observées sur les métriques `CurrConnections`. Ce comportement peut entraîner une saturation du ElastiCache côté ou du client. Lorsqu'une solution immédiate n'est pas possible du côté client, certains administrateurs définissent une valeur de « délai d'attente » dans leur groupe de ElastiCache paramètres. Le délai d'expiration est le temps, en secondes, autorisé pour que les connexions inactives persistent. Si le client ne soumet aucune demande pendant cette période, le moteur mettra fin à la connexion dès que la connexion atteindra le délai d'expiration. De petites valeurs de délai d'expiration peuvent entraîner des déconnexions inutiles et les clients devront les gérer correctement et se reconnecter, ce qui entraîne des retards.
- La mémoire utilisée pour stocker les clés est partagée avec les tampons client. Les clients lents avec des demandes ou des réponses importantes peuvent exiger une quantité importante de mémoire pour gérer ses tampons. La valeur par défaut ElastiCache pour les configurations Redis OSS ne limite pas la taille des tampons de sortie clients ordinaires. Si la limite `maxmemory` est atteinte, le moteur essaiera d'expulser les éléments pour remplir l'utilisation de la mémoire tampon. Dans des conditions de mémoire extrêmement limitées, ElastiCache Redis OSS peut choisir de

déconnecter les clients qui consomment de grandes quantités de mémoire tampon de sortie afin de libérer de la mémoire et de préserver l'intégrité du cluster.

Il est possible de limiter la taille des tampons client avec des configurations personnalisées et les clients atteignant la limite seront déconnectés. Cependant, les clients devraient être en mesure de gérer des déconnexions inattendues. Les paramètres permettant de gérer la taille des tampons pour les clients réguliers sont les suivants :

- client-query-buffer-limit: taille maximale d'une seule demande d'entrée ;
- client-output-buffer-limit- normal-soft-limit : Limite souple pour les connexions client. La connexion sera interrompue si elle reste au-dessus de la limite souple pendant plus de temps que le temps en secondes défini sur client-output-buffer-limit - normal-soft-seconds ou si elle atteint la limite stricte ;
- client-output-buffer-limit- normal-soft-seconds : Durée autorisée pour les connexions dépassant le client-output-buffer-limit - normal-soft-limit ;
- client-output-buffer-limit- normal-hard-limit : Une connexion atteignant cette limite sera immédiatement interrompue.

Outre les tampons clients classiques, les options suivantes contrôlent la mémoire tampon pour les nœuds de réplication et les clients Pub/Sub (Publish/Subscribe) :

- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-replica-soft-seconds;
- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-pubsub-soft-limit;
- client-output-buffer-limit-pubsub-soft-seconds;
- client-output-buffer-limit-pubsub-hard-limit;

Résolution des problèmes côté client pour les instances Amazon EC2

La charge et la réactivité du côté client peuvent également affecter les demandes adressées à ElastiCache. EC2 les limites des instances et du système d'exploitation doivent être soigneusement examinées lors de la résolution des problèmes de connectivité intermittents ou de temporisation. Quelques points clés à observer :

- UC :

- EC2 utilisation du processeur de l'instance : assurez-vous que le processeur n'est pas saturé ou qu'il n'est pas presque saturé à 100 %. L'analyse historique peut être effectuée via CloudWatch, mais gardez à l'esprit que la granularité des points de données est de 1 minute (avec la surveillance détaillée activée) ou de 5 minutes ;
- Si vous utilisez des EC2 instances burstable, assurez-vous que le solde créditeur de leur processeur n'a pas été épuisé. Ces informations sont disponibles sur la `CPUCreditBalance` CloudWatch métrique.
- De courtes périodes d'utilisation élevée du processeur peuvent entraîner des délais d'attente sans que cela ne se répercute sur une utilisation à 100 %. CloudWatch De tels cas nécessitent une surveillance en temps réel avec des outils de système d'exploitation tels que `top`, `ps` et `mpstat`.

- Réseau
  - Vérifiez si le débit réseau est inférieur à des valeurs acceptables en fonction des capacités de l'instance. Pour plus d'informations, consultez Amazon EC2 Instance Types
  - Sur les instances avec le pilote réseau amélioré `ena`, cochez la case ena statistics (statistiques ena) pour les délais d'expiration ou les limites dépassés. Les statistiques suivantes sont utiles pour confirmer la saturation des limites réseau :
    - `bw_in_allowance_exceeded` / `bw_out_allowance_exceeded` : nombre de paquets formés en raison d'un débit entrant ou sortant excessif ;
    - `conntrack_allowance_exceeded` : nombre de paquets abandonnés en raison de limites de suivi de connexion des groupes de sécurité. Les nouvelles connexions échouent lorsque cette limite est saturée ;
    - `linklocal_allowance_exceeded` : nombre de paquets abandonnés en raison de requêtes excessives vers des métadonnées d'instance, NTP via VPC DNS. La limite est de 1 024 paquets par seconde pour tous les services ;
    - `pps_allowance_exceeded` : nombre de paquets abandonnés en raison d'un rapport excessif de paquets par seconde. La limite PPS peut être atteinte lorsque le trafic réseau consiste en des milliers ou des millions de très petites requêtes par seconde. ElastiCache le trafic peut être optimisé pour mieux utiliser les paquets réseau via des pipelines ou des commandes qui effectuent plusieurs opérations à la fois, par exemple `MGET` au lieu de`GET`.

Disséquer le temps nécessaire pour compléter une seule demande

- Sur le réseau : `Tcpdump` et `Wireshark` (tshark sur la ligne de commande) sont des outils pratiques qui permettent de comprendre combien de temps il a fallu à la demande pour parcourir le réseau, atteindre le ElastiCache moteur et obtenir un retour. L'exemple suivant met en évidence une demande unique créée à l'aide de la commande suivante :

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

Parallèlement à la commande ci-dessus, tcpdump était en exécution et retourné :

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
 8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
 53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
 length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
 options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
 options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
 7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
 [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
```

```
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

À partir de la sortie ci-dessus, nous pouvons confirmer que la connexion en trois temps TCP a été terminée en 222 microsecondes (918 091 - 917 869) et que la commande ping a été envoyée et retournée en 173 microsecondes (918 295 - 918 122).

Il a fallu 438 microsecondes (918 307 - 917 869) entre la demande et la fermeture de la connexion. Ces résultats confirmeraient que les temps de réponse du réseau et du moteur sont bons et que l'enquête peut se concentrer sur d'autres composants.

- Sur le système d'exploitation : `Strace` peut contribuer à identifier les écarts de temps au niveau du système d'exploitation. L'analyse des applications réelles serait beaucoup plus étendue et des profileurs d'applications ou des débogueurs spécialisés sont conseillés. L'exemple suivant montre simplement si les composants du système d'exploitation de base fonctionnent comme prévu, sinon une enquête plus approfondie peut s'avérer nécessaire. En utilisant la même `PING` commande Redis OSS, `strace` nous obtenons :

```
$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
 nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)
    6379
1609430221.697712 (+ 0.000000) execve("/usr/bin/nc", ["nc",
 "example.xxxxxx.ng.0001.use"..., "6379"], 0x7fffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
    (+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
    (+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
 IPPROTO_IP) = 3
1609430221.717419
    (+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\rnotls20201224\6tihew"..., 2048, 0, {sa_family=AF_INET,
 sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
 "\204\207\201\200\0\1\0\1\0\0\0\0\rnotls20201224\6tihew"...,
```

```
    65536, 0, {sa_family=AF_INET, sin_port=htons(53),
 sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
    0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
 [128]) = -1 ENOTSOCK
    (Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
 [128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
    (+ 0.003569) +++ exited with 0 +++
```

Dans l'exemple ci-dessus, la commande a pris un peu plus de 54 millisecondes (752 110 - 697 712 = 54 398 microsecondes).

Un temps important, environ 20 ms, était nécessaire pour instancier nc et effectuer la résolution de nom (de 697 712 à 717 890). Ensuite, 2 ms ont été nécessaires pour créer le socket TCP (745 659 à 747 858), et 0,4 ms (747 858 à 748 330) pour envoyer et recevoir la réponse à la demande.

Rubriques connexes

- the section called "Bonnes pratiques et stratégies de mise en cache"

Sécurité sur Amazon ElastiCache

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 cette notion par les termes sécurité du cloud et 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 conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon ElastiCache, consultez la section AWS Services concernés par programme de conformité.
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon ElastiCache. Les rubriques suivantes expliquent comment configurer Amazon pour répondre ElastiCache 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 ElastiCache ressources Amazon.

Rubriques

- Protection des données sur Amazon ElastiCache
- Confidentialité du trafic inter-réseau
- Identity and Access Management pour Amazon ElastiCache
- Validation de conformité pour Amazon ElastiCache
- Résilience chez Amazon ElastiCache
- Sécurité de l'infrastructure dans AWS ElastiCache
- Mises à jour du service dans ElastiCache
- Vulnérabilités et expositions courantes (CVE) : vulnérabilités de sécurité corrigées dans ElastiCache

Protection des données sur Amazon ElastiCache

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS ElastiCache (ElastiCache). Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure mondiale qui gère l'ensemble du AWS cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu inclut la configuration de la sécurité et les tâches de gestion pour les AWS services que vous utilisez. Pour plus d'informations sur la confidentialité des données, veuillez consulter la FAQ sur la confidentialité des données.

Pour des raisons de protection des données, nous vous recommandons de protéger les informations d'identification des AWS comptes et de configurer des comptes individuels avec 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 TLS pour communiquer avec les AWS ressources.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut au sein AWS des services.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.

Nous vous recommandons vivement de ne jamais placer d'informations identifiables sensibles, telles que les numéros de compte de vos clients, dans des champs de formulaire comme Nom. Cela inclut lorsque vous travaillez avec ElastiCache ou avec d'autres AWS services à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans ElastiCache ou d'autres services peuvent être récupérées pour être insérées dans des journaux de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Rubriques

- Sécurité des données sur Amazon ElastiCache

Sécurité des données sur Amazon ElastiCache

Pour garantir la sécurité de vos données, Amazon ElastiCache et Amazon EC2 fournissent des mécanismes de protection contre tout accès non autorisé à vos données sur le serveur.

Amazon ElastiCache for Memcached fournit des fonctionnalités de chiffrement pour les données contenues dans les caches exécutant les versions 1.6.12 ou ultérieures de Memcached.

Amazon ElastiCache avec Valkey et Redis OSS fournit des fonctionnalités de chiffrement pour les données des caches exécutant Valkey 7.2 ou version ultérieure, et Redis OSS versions 3.2.6 (date de fin de vie prévue, voir calendrier de fin de vie des versions Redis OSS), 4.0.10 ou versions ultérieures. Amazon prend ElastiCache également en charge l'authentification des utilisateurs avec IAM ou Valkey et Redis OSS AUTH, et l'autorisation des opérations des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC).

- 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 cache et votre application.
- Le chiffrement au repos chiffre vos données sur disque au cours de la synchronisation et des opérations de sauvegarde.

ElastiCache prend en charge l'authentification des utilisateurs à l'aide d'IAM et de la commande Valkey et Redis OSS AUTH, et l'autorisation des opérations des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC).

ElastiCache pour le schéma de sécurité de Valkey et Redis OSS

Rubriques

- ElastiCache chiffrement en transit (TLS)
- Chiffrement au repos dans ElastiCache
- Authentification et autorisation

ElastiCache chiffrement en transit (TLS)

Pour garantir la sécurité de vos données, Amazon ElastiCache 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, vous ElastiCache disposez d'un outil que vous pouvez utiliser pour protéger vos données lorsqu'elles sont déplacées d'un endroit à un autre.

Le chiffrement en transit est activé dans tous les caches sans serveur Valkey ou Redis OSS. Pour les clusters auto-conçus, vous pouvez activer le chiffrement en transit sur un groupe de réplication en définissant le paramètre `TransitEncryptionEnabled` sur `true` (interface de ligne de commande : `--transit-encryption-enabled`) lorsque vous créez le groupe de réplication. Vous pouvez le faire, que vous créiez le groupe de réplication à l'aide de AWS Management Console, de AWS CLI, ou de l' ElastiCache API.

Le chiffrement en transit est activé pour tous les caches sans serveur. Pour les clusters auto-conçus, vous pouvez activer le chiffrement en transit sur un cluster de cache en définissant le paramètre `TransitEncryptionEnabled` sur `true` (interface de ligne de commande : `--transit-encryption-enabled`) lorsque vous créez le cluster de cache à l'aide de l'opération `CreateCacheCluster` (interface de ligne de commande : `create-cache-cluster`).

Rubriques

- Présentation du chiffrement en transit
- Conditions de chiffrement en transit (Valkey et Redis OSS)
- Conditions de chiffrement en cours de transit (Memcached)
- Bonnes pratiques en matière de chiffrement en transit
- Autres options Valkey et Redis OSS
- Activation du chiffrement en transit pour Memcached
- Activation du chiffrement en transit
- Connexion à ElastiCache (Valkey) ou Amazon ElastiCache pour Redis OSS avec chiffrement en transit à l'aide de valkey-cli
- Activation du chiffrement en transit sur un cluster Redis OSS conçu par ses soins à l'aide de Python
- Bonnes pratiques lors de l'activation du chiffrement en transit
- Connexion à des nœuds activés avec le chiffrement en transit à l'aide d'OpenSSL (Memcached)
- Création d'un client Memcached en mode TLS avec Java
- Création d'un client Memcached en mode TLS avec PHP

Présentation du chiffrement en transit

Le chiffrement ElastiCache en transit d'Amazon est une fonctionnalité qui vous permet de renforcer la sécurité de vos données aux points les plus vulnérables, lorsqu'elles sont en transit d'un endroit à un autre. Puisque certains traitements nécessitaient de chiffrer et de déchiffrer les données aux points de terminaison, l'activation du chiffrement en transit peut avoir un certain impact sur la performance. Vous devez référencer vos données avec et sans le chiffrement en transit pour déterminer l'impact sur la performance pour vos cas d'utilisation.

ElastiCache le chiffrement en transit met en œuvre les fonctionnalités suivantes :

- Connexions client chiffrées : les connexions client aux nœuds de cache sont chiffrées à l'aide du protocole TLS.
- Connexions serveur chiffrées : les données transférées entre les nœuds d'un cluster sont chiffrées.
- Authentification du serveur : les clients peuvent authentifier leur connexion au bon serveur.
- Authentification du client : à l'aide des fonctionnalités Valkey et Redis OSS AUTH, le serveur peut authentifier les clients.

Conditions de chiffrement en transit (Valkey et Redis OSS)

Les contraintes suivantes relatives au chiffrement ElastiCache en transit d'Amazon doivent être prises en compte lorsque vous planifiez la mise en œuvre de votre propre cluster :

- Le chiffrement en transit est pris en charge sur les groupes de réplication exécutant Valkey 7.2 et versions ultérieures, et Redis OSS versions 3.2.6, 4.0.10 et ultérieures.
- La modification du paramètre de chiffrement en transit, pour un cluster existant, est prise en charge sur les groupes de réplication exécutant Valkey 7.2 et versions ultérieures, et Redis OSS version 7 et ultérieure.
- Le chiffrement en transit est pris en charge uniquement par les groupes de réplication exécutés dans un Amazon VPC.
- Le chiffrement en transit n'est pas pris en charge pour les groupes de réplication exécutant les types de nœuds suivants : M1, M2.

  Pour de plus amples informations, veuillez consulter Types de nœuds pris en charge.
- Le chiffrement en transit est activé en définissant explicitement le paramètre `TransitEncryptionEnabled` sur `true`.
- Assurez-vous que votre client de mise en cache prend en charge la connectivité TLS et que vous l'avez activée dans la configuration du client.
- À compter du 26 janvier 2026, la version TLS minimale prise en charge AWS sera mise à jour vers la version 1.2 ElastiCache pour Valkey version 7.2 et supérieure, et ElastiCache pour Redis OSS version 6 et supérieure. Les clients doivent mettre à jour leur logiciel client avant cette date. Cette mise à jour vous aide à répondre aux besoins en matière de sécurité, de conformité et de réglementation.

Conditions de chiffrement en cours de transit (Memcached)

Les contraintes suivantes relatives au chiffrement ElastiCache en transit d'Amazon doivent être prises en compte lorsque vous planifiez la mise en œuvre de votre propre cluster :

- Le chiffrement en transit est pris en charge sur les clusters exécutant Memcached versions 1.6.12 et ultérieures.
- Le chiffrement en transit prend en charge le protocole TLS (Transport Layer Security) versions 1.2 et 1.3.
- Le chiffrement en transit est pris en charge uniquement par les clusters exécutés dans un Amazon VPC.
- Le chiffrement en transit n'est pas pris en charge pour les groupes de réplication exécutant les types de nœuds suivants : M1, M2, M3, R3, T2.

  Pour de plus amples informations, veuillez consulter Types de nœuds pris en charge.
- Le chiffrement en transit est activé en définissant explicitement le paramètre `TransitEncryptionEnabled` sur `true`.
- Vous pouvez activer le chiffrement en transit sur un cluster uniquement lors de la création du cluster. Vous ne pouvez pas activer et désactiver le chiffrement en transit en modifiant un cluster.
- Assurez-vous que votre client de mise en cache prend en charge la connectivité TLS et que vous l'avez activée dans la configuration du client.

Bonnes pratiques en matière de chiffrement en transit

- Puisque le traitement nécessitait de chiffrer et de déchiffrer les données aux points de terminaison, l'implémentation du chiffrement en transit peut réduire la performance. Référencez le chiffrement en transit en comparaison avec l'absence de chiffrement sur vos propres données pour déterminer l'impact sur la performance pour votre cas d'utilisation.
- La création de nouvelles connexions étant parfois coûteuse, vous pouvez réduire l'impact du chiffrement en transit sur les performances en conservant vos connexions TLS.

Autres options Valkey et Redis OSS

Pour plus d'informations sur les options disponibles pour Valkey et Redis OSS, consultez les liens suivants.

- Chiffrement au repos dans ElastiCache

- Authentification avec la commande Valkey et Redis OSS AUTH
- Contrôle d'accès basé sur les rôles (RBAC)
- Amazon VPCs et la ElastiCache sécurité
- Identity and Access Management pour Amazon ElastiCache

Activation du chiffrement en transit pour Memcached

Pour activer le chiffrement en transit lors de la création d'un cluster Memcached à l'aide de la Console de gestion AWS , effectuez les sélections suivantes :

- Choisissez Memcached comme moteur.
- Choisissez la version de moteur 1.6.12 ou ultérieure.
- Sous Encryption in transit (Chiffrement en transit), choisissez Enable (Activer).

Pour le step-by-step processus, voirCréation d'un cluster pour Valkey ou Redis OSS.

Activation du chiffrement en transit

Le chiffrement en transit est activé pour tous les caches sans serveur. Sur un cluster conçu par vos soins, vous pouvez activer le chiffrement en transit à l'aide de la AWS Management Console AWS CLI ou de l' ElastiCacheAPI.

Activation du chiffrement en transit à l'aide du AWS Management Console

Activation du chiffrement en transit pour un nouveau cluster auto-conçu à l'aide du AWS Management Console

Lorsque vous concevez votre propre cluster, le chiffrement en transit est activé pour les configurations « Dev/Test » et « Production » avec la méthode « Création facile ». Lorsque vous choisissez vous-même la configuration, effectuez les sélections suivantes :

- Choisissez la version de moteur 3.2.6, 4.0.10 ou ultérieure.
- Cochez la case en regard de Activer pour l'option Chiffrement en transit.

Pour le step-by-step processus, consultez ce qui suit :

- Création d'un cluster Valkey (mode cluster désactivé) (console)
- Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Activation du chiffrement en transit pour un cluster autoconçu existant à l'aide du AWS Management Console

L'activation du chiffrement en transit est un processus en deux étapes. Vous devez d'abord définir le mode de chiffrement en transit sur `preferred`. Ce mode permet à vos clients Valkey ou Redis OSS de se connecter en utilisant des connexions chiffrées et non chiffrées. Après avoir migré tous vos clients Valkey ou Redis OSS pour utiliser des connexions chiffrées, vous pouvez modifier la configuration de votre cluster pour définir le mode de cryptage du transit sur. `required` Définir le mode de chiffrement en transit sur `required` supprimera toutes les connexions non chiffrées et n'autorisera que les connexions chiffrées.

Définissez votre mode de cryptage Transit sur Préféré

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Choisissez les caches Valkey ou les caches Redis OSS dans les ElastiCache ressources répertoriées dans le volet de navigation, sur la gauche.
3. Choisissez le cache que vous souhaitez mettre à jour.
4. Choisissez la liste déroulante Actions, puis Modify (Modifier).
5. Choisissez Enable (Activer) sous Encryption in transit (Chiffrement en transit) dans la section Security (Sécurité).
6. Choisissez Preferred (Préféré) comme Transit encryption mode (Mode de chiffrement en transit).
7. Choisissez Preview changes (Prévisualiser les modifications), puis enregistrez vos modifications.

Après avoir migré tous vos clients Valkey ou Redis OSS pour utiliser des connexions chiffrées :

Définissez le mode de cryptage de votre Transit sur Obligatoire

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Choisissez les caches Valkey ou les caches Redis OSS dans les ElastiCache ressources répertoriées dans le volet de navigation, sur la gauche.
3. Choisissez le cache que vous souhaitez mettre à jour.
4. Choisissez la liste déroulante Actions, puis Modify (Modifier).
5. Choisissez Required (Obligatoire) comme Transit encryption mode (Mode de chiffrement en transit) dans la section Security (Sécurité).

6. Choisissez Preview changes (Prévisualiser les modifications), puis enregistrez vos modifications.

Activation du chiffrement en transit à l'aide du AWS CLI

Pour activer le chiffrement en transit lors de la création d'un groupe de réplication Valkey ou Redis OSS à l'aide du paramètre AWS CLI, utilisez le paramètre. `transit-encryption-enabled`

Activation du chiffrement en transit sur un nouveau cluster auto-conçu pour Valkey ou Redis OSS (mode cluster désactivé) (CLI)

Utilisez l' AWS CLI opération `create-replication-group` et les paramètres suivants pour créer un groupe de réplication Valkey ou Redis OSS avec des répliques pour lesquelles le chiffrement en transit est activé :

Paramètres clés :

- **`--engine`**—Ça doit être `valkey` ou. `redis`
- **`--engine-version`**—Si le moteur est Redis OSS, il doit être 3.2.6, 4.0.10 ou version ultérieure.
- **`--transit-encryption-enabled`** : obligatoire. Si vous activez le chiffrement en transit, vous devez également fournir une valeur pour le paramètre `--cache-subnet-group`.
- **`--num-cache-clusters`** : 1 au minimum. La valeur maximale pour ce paramètre est 6.

Pour plus d'informations, consultez les ressources suivantes :

- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro ()AWS CLI
- create-replication-group

Activation du chiffrement en transit sur un nouveau cluster auto-conçu pour Valkey ou Redis OSS (mode cluster activé) (CLI)

Utilisez l' AWS CLI opération `create-replication-group` et les paramètres suivants pour créer un groupe de réplication Valkey ou Redis OSS (mode cluster activé) sur lequel le chiffrement en transit est activé :

Paramètres clés :

- **`--engine`**—Ça doit être `valkey` ou. `redis`

- **`--engine-version`**—Si le moteur est Redis OSS, il doit être 3.2.6, 4.0.10 ou version ultérieure.
- **`--transit-encryption-enabled`** : obligatoire. Si vous activez le chiffrement en transit, vous devez également fournir une valeur pour le paramètre `--cache-subnet-group`.
- Pour spécifier la configuration des groupes de nœud du groupe de réplication, utilisez l'un des ensembles de paramètres suivants :
  - **`--num-node-groups`** : spécifie le nombre de partitions (groupes de nœuds) de ce groupe de réplication. La valeur maximale pour ce paramètre est 500.

    `--replicas-per-node-group` : spécifie le nombre de nœuds de réplicas dans chaque groupe de nœuds. La valeur spécifiée ici est appliquée à toutes les partitions de ce groupe de réplication. La valeur maximale pour ce paramètre est 5.
  - **`--node-group-configuration`** : spécifie la configuration de chaque partition de manière indépendante.

Pour plus d'informations, consultez les ressources suivantes :

- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro ()AWS CLI
- create-replication-group

Activation du chiffrement en transit pour un cluster existant à l'aide de la AWS CLI

L'activation du chiffrement en transit est un processus en deux étapes. Vous devez d'abord définir le mode de chiffrement en transit sur `preferred`. Ce mode permet à vos clients Valkey ou Redis OSS de se connecter en utilisant des connexions chiffrées et non chiffrées. Après avoir migré tous vos clients Valkey ou Redis OSS pour utiliser des connexions chiffrées, vous pouvez modifier la configuration de votre cluster pour définir le mode de cryptage du transit sur. `required` Définir le mode de chiffrement en transit sur `required` supprimera toutes les connexions non chiffrées et n'autorisera que les connexions chiffrées.

Utilisez l' AWS CLI opération `modify-replication-group` et les paramètres suivants pour mettre à jour un groupe de réplication Valkey ou Redis OSS (mode cluster activé) pour lequel le chiffrement en transit est désactivé.

Pour activer le chiffrement en transit

1. transit-encryption-modeRéglé sur`preferred`, en utilisant les paramètres suivants

- **`--transit-encryption-enabled`** : obligatoire.
- **`--transit-encryption-mode`** : doit être défini sur `preferred`.

2. Réglez transit-encryption-mode sur`required`, en utilisant les paramètres suivants :

- **`--transit-encryption-enabled`** : obligatoire.
- **`--transit-encryption-mode`** : doit être défini sur `required`.

Connexion à ElastiCache (Valkey) ou Amazon ElastiCache pour Redis OSS avec chiffrement en transit à l'aide de valkey-cli

Pour accéder aux données depuis ElastiCache les caches Redis OSS activés avec le chiffrement en transit, vous utilisez des clients qui fonctionnent avec le protocole SSL (Secure Socket Layer). Vous pouvez également utiliser valkey-cli sur TLS/SSL Amazon Linux et Amazon Linux 2. Si votre client ne prend pas en charge le protocole TLS, vous pouvez utiliser la `stunnel` commande sur l'hôte de votre client pour créer un tunnel SSL vers les nœuds Redis OSS.

Connexion cryptée avec Linux

Pour utiliser valkey-cli pour vous connecter à un cluster Valkey ou Redis OSS activé avec le chiffrement en transit sur Amazon Linux 2 ou Amazon Linux, procédez comme suit.

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

   Amazon Linux 2

   Si vous utilisez Amazon Linux 2, entrez ceci :

```
sudo yum -y install openssl-devel gcc
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean
make valkey-cli BUILD_TLS=yes
sudo install -m 755 src/valkey-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 https://github.com/valkey-io/valkey/archive/refs/tags/8.0.0.tar.gz
tar xvzf valkey-8.0.0.tar.gz
cd valkey-8.0.0
make valkey-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/valkey-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 valkey-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 :

```
valkey-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 : `valkey-cli`
>
> ```
> redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
> ```

Connexion cryptée avec Stunnel

Pour utiliser valkey-cli pour vous connecter à un cluster Redis OSS activé avec le chiffrement en transit à l'aide de Stunnel, procédez comme suit.

1. Utilisez SSH pour vous connecter à votre client et installer `stunnel`.

```
sudo yum install stunnel
```

2. Exécutez la commande suivante pour créer et modifier le fichier `'/etc/stunnel/valkey-cli.conf'` simultanément afin d'ajouter un ElastiCache point de terminaison du cluster Redis OSS à un ou plusieurs paramètres de connexion, en utilisant le résultat fourni ci-dessous comme modèle.

```
vi /etc/stunnel/valkey-cli.conf


fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[valkey-cli]
   client = yes
   accept = 127.0.0.1:6379
   connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-cli-replica]
   client = yes
   accept = 127.0.0.1:6380
   connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

Dans cet exemple, le fichier de configuration comporte deux connexions : `valkey-cli` et `valkey-cli-replica`. Les paramètres sont définis comme suit :

- client est défini sur Yes (Oui) pour indiquer que cette instance stunnel est un client.
- accept est défini sur l'adresse IP du client. Dans cet exemple, le serveur principal est défini sur la valeur par défaut de Redis OSS 127.0.0.1 sur le port 6379. Le réplica doit appeler un port différent et défini sur 6380. Vous pouvez utiliser les ports éphémères 1024–65535. Pour de plus amples informations, veuillez consulter Ports éphémères dans le Guide de l'utilisateur Amazon VPC.
- connect est défini sur le point de terminaison du serveur Redis OSS. Pour de plus amples informations, veuillez consulter Recherche de points de terminaison de connexion dans ElastiCache.

3. Démarrer `stunnel`.

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

Utilisez la commande `netstat` pour confirmer que les tunnels ont démarré.

```
sudo netstat -tulnp | grep -i stunnel

tcp        0      0 127.0.0.1:6379              0.0.0.0:*                   LISTEN
     3189/stunnel
tcp        0      0 127.0.0.1:6380              0.0.0.0:*                   LISTEN
     3189/stunnel
```

4. Connectez-vous au nœud Redis OSS crypté à l'aide du point de terminaison local du tunnel.

   - Si aucun mot de passe AUTH n'a été utilisé lors ElastiCache de la création du cluster Redis OSS, cet exemple utilise le valkey-cli pour se connecter au serveur Redis OSS en utilisant le chemin complet ElastiCache pour valkey-cli, sur Amazon Linux :

     ```
     /home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
     ```

     Si le mot de passe AUTH a été utilisé lors de la création du cluster Redis OSS, cet exemple utilise valkey-cli pour se connecter au serveur Redis OSS en utilisant le chemin complet pour valkey-cli, sur Amazon Linux :

     ```
      /home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-
     password
     ```

   OU

   - Changez le répertoire en redis-7.2.5 et procédez comme suit :

     Si aucun mot de passe AUTH n'a été utilisé lors ElastiCache de la création du cluster Redis OSS, cet exemple utilise le valkey-cli pour se connecter au serveur Redis OSS en utilisant le chemin complet ElastiCache pour valkey-cli, sur Amazon Linux :

     ```
     src/valkey-cli -h localhost -p 6379
     ```

Si le mot de passe AUTH a été utilisé lors de la création du cluster Redis OSS, cet exemple utilise valkey-cli pour se connecter au serveur Valkey ou Redis OSS en utilisant le chemin complet pour valkey-cli, sur Amazon Linux :

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

Cet exemple utilise Telnet pour se connecter au serveur Valkey Redis OSS.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Pour arrêter et fermer les tunnels SSL, arrêtez le processus stunnel `pkill`.

```
sudo pkill stunnel
```

Activation du chiffrement en transit sur un cluster Redis OSS conçu par ses soins à l'aide de Python

Le guide suivant explique comment activer le chiffrement en transit sur un cluster Redis OSS 7.0 créé à l'origine avec le chiffrement en transit désactivé. Les clients TCP et TLS continuent de communiquer avec le cluster pendant ce processus sans interruption de service.

Boto3 obtient les informations d'identification nécessaires (`aws_access_key_id`, `aws_secret_access_key` et `aws_session_token`) à partir des variables d'environnement. Ces informations d'identification sont collées à l'avance dans le même terminal bash où nous exécutons `python3` pour traiter le code Python présenté dans ce guide. Le code de l'exemple ci-dessous a été traité à partir d'une EC2 instance lancée dans le même VPC qui sera utilisé pour y créer le cluster ElastiCache Redis OSS.

> **Note**
>
> - Les exemples suivants utilisent le SDK boto3 pour les opérations de ElastiCache gestion (création de clusters ou d'utilisateurs) et redis-py-cluster redis-py/ pour le traitement des données.
> - Vous devez utiliser au moins la version Boto3 (=~) 1.26.39 pour utiliser la migration TLS en ligne avec l'API de modification du cluster.
> - ElastiCache prend en charge la migration TLS en ligne uniquement pour les clusters dotés de Valkey version 7.2 ou supérieure ou de Redis OSS version 7.0 ou supérieure. Ainsi, si vous avez un cluster exécutant une version de Redis OSS antérieure à 7.0, vous devrez mettre à niveau la version Redis OSS de votre cluster. Pour plus d'informations sur ces différences de version, consultez Principales différences de comportement entre les versions du moteur et de compatibilité avec Redis OSS.

Rubriques

- Définissez les constantes de chaîne qui lanceront le cluster ElastiCache Valkey ou Redis OSS
- Définir les classes pour la configuration du cluster
- Définir une classe qui représente le cluster lui-même
- (Facultatif) Créez une classe wrapper pour démontrer la connexion du client au cluster Valkey ou Redis OSS
- Créer la fonction principale qui montre le processus de modification de la configuration du chiffrement en transit

Définissez les constantes de chaîne qui lanceront le cluster ElastiCache Valkey ou Redis OSS

Définissons d'abord quelques constantes de chaîne Python simples qui contiendront les noms des AWS entités requises pour créer le ElastiCache cluster`security-group`, telles que`Cache Subnet group`, et `default parameter group` a. Toutes ces AWS entités doivent être créées à l'avance sur votre AWS compte dans la région que vous souhaitez utiliser.

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
 migration user guide"
EC_SUBNET_GROUP = "client-testing"
```

```
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

Définir les classes pour la configuration du cluster

Définissons maintenant quelques classes Python simples qui représenteront la configuration d'un cluster, qui contiendra des métadonnées sur le cluster, telles que la version de Valkey ou Redis OSS, le type d'instance et si le chiffrement en transit (TLS) est activé ou désactivé.

```
#Class definitions

class Config:
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
    ):
        self.instance_type = instance_type
        self.version = version
        self.multi_az = multi_az
        self.TLS = TLS
        self.name = name or f"tls-test"

    def create_base_launch_request(self):
        return {
            "ReplicationGroupId": self.name,
            "TransitEncryptionEnabled": self.TLS,
            "MultiAZEnabled": self.multi_az,
            "CacheNodeType": self.instance_type,
            "Engine": "redis",
            "EngineVersion": self.version,
            "CacheSubnetGroupName": EC_SUBNET_GROUP ,
            "CacheParameterGroupName":
 DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
            "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
            "SecurityGroupIds": [SECURITY_GROUP],
        }

class ConfigCME(Config):
    def __init__(
        self,
```

```
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
        num_shards: int = 2,
        num_replicas_per_shard: int = 1,
    ):
        super().__init__(instance_type, version, multi_az, TLS, name)
        self.num_shards = num_shards
        self.num_replicas_per_shard = num_replicas_per_shard

    def create_launch_request(self) -> dict:
        launch_request = self.create_base_launch_request()
        launch_request["NumNodeGroups"] = self.num_shards
        launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
        return launch_request
```

Définir une classe qui représente le cluster lui-même

Définissons maintenant quelques classes Python simples qui représenteront le cluster ElastiCache Valkey ou Redis OSS lui-même. Cette classe aura un champ client qui contiendra un client boto3 pour les opérations de ElastiCache gestion telles que la création du cluster et l'interrogation de l'API. ElastiCache

```
import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)


class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()
```

```
    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(
        ReplicationGroupId=self.name
        )["ReplicationGroups"][0]

    def get_status(self) -> str:
        return self.get_first_replication_group()["Status"]

    def get_transit_encryption_enabled(self) -> bool:
        return self.get_first_replication_group()["TransitEncryptionEnabled"]

    def is_available(self) -> bool:
        return self.get_status() == "available"

    def is_modifying(self) -> bool:
        return self.get_status() == "modifying"

    def wait_for_available(self):
        while True:
            if self.is_available():
                break
            else:
                time.sleep(5)

    def wait_for_modifying(self):
        while True:
            if self.is_modifying():
                break
            else:
                time.sleep(5)

    def delete_cluster(self) -> bool:
        self.elasticache_client.delete_replication_group(
            ReplicationGroupId=self.name, RetainPrimaryCluster=False
        )

    def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
        # generate api call to migrate the cluster to TLS preffered or to TLS required
            self.elasticache_client.modify_replication_group(
                ReplicationGroupId=self.name,
                TransitEncryptionMode=new_transit_encryption_mode,
                TransitEncryptionEnabled=True,
                ApplyImmediately=True,
            )
```

```
        self.wait_for_modifying()

 class ElastiCacheClusterCME(ElastiCacheClusterBase):
    def __init__(self, name: str):
        super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
 readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()
```

(Facultatif) Créez une classe wrapper pour démontrer la connexion du client au cluster Valkey ou Redis OSS

Créons maintenant une classe encapsuleur pour le client `redis-py-cluster`. Cette classe encapsuleur permet de pré-remplir le cluster avec certaines clés, puis d'exécuter des commandes `get` répétées de manière aléatoire.

> **Note**
>
> Il s'agit d'une étape facultative, mais elle simplifie le code de la fonction principale qui sera ajoutée à une étape ultérieure.

```
import redis
improt random
from time import perf_counter_ns, time


class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
 prefilled
        # total downtime in nano seconds
        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0


    def replace_client(self, client):
        self.client = client

    def prefill_data(self, timelimit_sec=60):
        end_time = time() + timelimit_sec
        while time() < end_time:
            self.client.set(self.prefilled, self.prefilled)
            self.prefilled += 1

    # unsuccesful operations throw exceptions
    def _exec(self, func):
        try:
            start_ns = perf_counter_ns()
```

```
            func()
            self.success_ops += 1
            elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
            # upon succesful execution of func
            # reset random_key to None so that the next command
            # will use a new random key
            self.random_key = None

        except Exception as e:
            elapsed_ns = perf_counter_ns() - start_ns
            self.downtime_ns += elapsed_ns
            # in case of failure- increment the relevant counters so that we will keep
track
            # of how many connection issues we had while trying to communicate with
            # the cluster.
            self.fail_ops += 1
            if e.__class__ is redis.exceptions.ConnectionError:
                self.connection_errors += 1
            if e.__class__ is redis.exceptions.TimeoutError:
                self.timeout_errors += 1

    def _repeat_exec(self, func, seconds):
        end_time = time() + seconds
        while time() < end_time:
            self._exec(func)

    def _new_random_key_if_needed(self, percent_above_prefilled):
        if self.random_key is None:
            max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
            return random.randint(0, max)
        return self.random_key

    def _random_get(self):
        key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
        result = self.client.get(key)
        # we know the key was set for sure only in the case key < self.prefilled
        if key < self.prefilled:
            assert result.decode("UTF-8") == str(key)


    def repeat_get(self, seconds=60):
        self._repeat_exec(self._random_get, seconds)

    def get_downtime_ms(self) -> int:
```

```
        return self.downtime_ns // 10 ** 6


    def do_get_until(self, cond_check):
        while not cond_check():
            self.repeat_get()
        # do one more get cycle once condition is met
        self.repeat_get()
```

Créer la fonction principale qui montre le processus de modification de la configuration du chiffrement en transit

Maintenant, définissons la fonction principale, qui effectue les opérations suivantes :

1. Créez le cluster à l'aide du client boto3 ElastiCache .
2. Initialisez le client `redis-py-cluster` qui se connecte au cluster avec une connexion TCP claire sans TLS.
3. Le client `redis-py-cluster` préremplit le cluster avec certaines données.
4. Le client Boto3 déclenche la migration TLS d'un protocole non compatible avec TLS vers un protocole TLS préféré.
5. Pendant la migration du cluster vers TLS `Preferred`, le client TCP `redis-py-cluster` envoie des opérations `get` répétées au cluster jusqu'à ce que la migration soit terminée.
6. Une fois la migration vers TLS `Preferred` terminée, nous affirmons que le cluster prend en charge le chiffrement en transit. Ensuite, nous créons un client `redis-py-cluster` qui se connecte au cluster via TLS.
7. Nous envoyons certaines commandes `get` en utilisant le nouveau client TLS et l'ancien client TCP.
8. Le client Boto3 déclenche la migration TLS d'un protocole TLS `Preferred` vers un protocole TLS obligatoire.
9. Lorsque la migration du cluster vers le protocole TLS est requise, le client redis-py-cluster TLS enverra des `get` opérations répétées au cluster jusqu'à ce que la migration soit terminée.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
 DowntimeTestClient:
```

```
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
 socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
    config = ConfigCME(TLS=False, instance_type="cache.m5.large")

    with ElastiCacheCMEManager(config) as cluster:
        # create a client that will connect to the cluster with clear tcp connection
        test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

       # migrate the cluster to TLS Preferred
        cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

        # do repeated get commands until the cluster finishes the migration to TLS
 Preferred
        test_client_tcp.do_get_until(cluster.is_available)

       # verify that in transit encryption is enabled so that clients will be able to
 connect to the cluster with TLS
        assert cluster.get_transit_encryption_enabled() == True

       # create a client that will connect to the cluster with TLS connection.
        # we must first make sure that the cluster indeed supports TLS
        test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

        # by doing get commands with the tcp client for 60 more seconds
       # we can verify that the existing tcp connection to the cluster still works
        test_client_tcp.repeat_get(seconds=60)

        # do get commands with the new TLS client for 60 more seconds
        test_client_tcp.repeat_get(seconds=60)

       # migrate the cluster to TLS required
        cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

       # from this point the tcp clients will be disconnected and we must not use them
 anymore.
```

```
      # do get commands with the TLS client until the cluster finishes migartion to
TLS required mode.
       test_client_tls.do_get_until(cluster.is_available)
```

Bonnes pratiques lors de l'activation du chiffrement en transit

Avant d'activer le chiffrement en transit : veillez à gérer correctement les enregistrements DNS

> **Note**
>
> Nous modifions et supprimons les anciens points de terminaison au cours de ce processus. Une utilisation incorrecte des points de terminaison peut amener le client Valkey ou Redis OSS à utiliser des points de terminaison anciens et supprimés, ce qui l'empêchera de se connecter au cluster.

Pendant la migration du cluster d'un point de terminaison No-TLS vers TLS préféré, l'ancien enregistrement DNS du point de terminaison de configuration du cluster est conservé et les nouveaux enregistrements DNS du point de terminaison de configuration du cluster sont générés dans un format différent. Les clusters compatibles TLS utilisent un format d'enregistrement DNS différent de celui des clusters activés par TLS. ElastiCache conservera les deux enregistrements DNS lorsqu'un cluster est configuré `encryption mode: Preferred` afin que les applications et les autres clients Valkey ou Redis OSS puissent passer de l'un à l'autre. Les modifications suivantes sont apportées aux enregistrements DNS au cours du processus de migration TLS :

Description des modifications apportées aux enregistrements DNS lors de l'activation du chiffrement en transit

Pour les clusters CME

Lorsqu'un cluster est réglé sur « mode de chiffrement en transit : préféré » :

- Le point de terminaison de configuration de cluster d'origine pour le cluster No-TLS restera actif. Il n'y a aucun temps d'arrêt lorsque le cluster est reconfiguré du mode de chiffrement TLS « sans » au mode « préféré ».
- De nouveaux points de terminaison TLS Valkey ou Redis OSS seront générés lorsque le cluster est défini sur le mode TLS préféré. Ces nouveaux points de terminaison seront résolus de la même manière IPs que les anciens (non TLS).

- Le nouveau point de terminaison de configuration TLS Valkey ou Redis OSS sera exposé dans la ElastiCache console et dans la réponse à l'API. `describe-replication-group`

Lorsqu'un cluster est réglé sur « mode de chiffrement en transit : obligatoire » :

- Les anciens points de terminaison non compatibles avec le protocole TLS sont supprimés. Les points de terminaison de cluster TLS ne sont pas interrompus.
- Vous pouvez en récupérer un nouveau `cluster-configuration-endpoint` depuis ElastiCache la console ou depuis l'`describe-replication-group`API.

Pour les clusters CMD avec basculement automatique activé ou désactivé

Lorsqu'un groupe de réplication est réglé sur « mode de chiffrement en transit : préféré » :

- Le point de terminaison principal d'origine et le point de terminaison du lecteur pour un cluster non compatible avec TLS restent actifs.
- De nouveaux points de terminaison principaux et du lecteur TLS sont générés lorsque le cluster est réglé sur le mode TLS `Preferred`. Ces nouveaux points de terminaison adoptent la ou les mêmes adresses IP que les anciens (non TLS).
- Le nouveau point de terminaison principal et le nouveau point de terminaison du lecteur seront exposés dans la ElastiCache console et dans la réponse à l'`describe-replication-group`API.

Lorsqu'un groupe de réplication est réglé sur « mode de chiffrement en transit : obligatoire » :

- Les anciens points de terminaison principaux et du lecteur non compatibles avec TLS sont supprimés. Les points de terminaison de cluster TLS ne sont pas interrompus.
- Vous pouvez récupérer les nouveaux points de terminaison principaux et lecteurs depuis ElastiCache la console ou depuis l'`describe-replication-group`API.

L'utilisation suggérée des enregistrements DNS

Pour les clusters CME

- Utilisez le point de terminaison de configuration du cluster plutôt que les enregistrements DNS par nœud dans le code de votre application. Il n'est pas recommandé d'utiliser directement les noms

DNS par nœud, car ils changeront au cours de la migration et le code de l'application interrompra la connexion au cluster.

- Ne codez pas en dur un point de terminaison de configuration de cluster dans votre application, car il changera au cours de ce processus.
- Coder en dur le point de terminaison de configuration du cluster dans votre application est une mauvaise pratique, car il peut être modifié au cours de ce processus. Une fois le chiffrement en transit terminé, interrogez le point de terminaison de configuration du cluster avec l'`describe-replication-group`API (comme illustré ci-dessus (en gras)) et utilisez le DNS que vous obtenez en réponse à partir de ce moment.

Pour les clusters CMD avec basculement automatique activé

- Utilisez le point de terminaison principal et le point de terminaison du lecteur plutôt que les noms DNS par nœud dans le code de votre application, car les anciens noms DNS par nœud sont supprimés et de nouveaux sont générés lors de la migration du cluster d'un protocole non compatible avec TLS vers TLS préféré. L'utilisation directe de noms DNS par nœud n'est pas recommandée si vous choisissez d'ajouter des réplicas à votre cluster à l'avenir. De plus, lorsque le basculement automatique est activé, les rôles du cluster principal et des répliques sont modifiés automatiquement par le ElastiCache service. Il est suggéré d'utiliser le point de terminaison principal et le point de terminaison du lecteur pour vous aider à suivre ces modifications. Enfin, l'utilisation du point de terminaison du lecteur vous permet de répartir vos lectures provenant des réplicas de manière égale entre les réplicas du cluster.
- Il est déconseillé de coder en dur le point de terminaison principal et le point de terminaison du lecteur dans votre application, car cela peut être modifié au cours du processus de migration TLS. Une fois la migration vers TLS-Preferred terminée, interrogez le point de terminaison principal et le point de terminaison du lecteur avec l' describe-replication-groupAPI et utilisez le DNS que vous obtenez en réponse à partir de ce moment. De cette façon, vous pouvez suivre l'évolution des points de terminaison de manière dynamique.

Pour les clusters CMD avec basculement automatique désactivé

- Utilisez le point de terminaison principal et le point de terminaison du lecteur plutôt que les noms DNS par nœud dans le code de votre application. Lorsque le basculement automatique est désactivé, c'est vous qui effectuez le dimensionnement, les correctifs, le basculement et les autres procédures gérées automatiquement par le ElastiCache service lorsque le basculement automatique est activé. Vous pouvez ainsi plus facilement effectuer un suivi manuel des différents

points de terminaison. Puisque les anciens noms DNS par nœud sont supprimés et que de nouveaux sont générés lors de la migration du cluster du mode non compatible avec TLS vers TLS préféré, n'utilisez pas directement les noms DNS par nœud. Cela est obligatoire pour que les clients puissent se connecter au cluster pendant la migration vers TLS. Vous bénéficiez également de la répartition uniforme des lectures entre les réplicas lorsque vous utilisez le point de terminaison du lecteur et du suivi des enregistrements DNS lors de l'ajout ou de la suppression de réplicas dans le cluster.

- Il est déconseillé de coder en dur le point de terminaison de configuration du cluster dans votre application car il peut être modifié au cours du processus de migration TLS.

Pendant le chiffrement en transit : faites attention à la fin du processus de migration

Le passage au mode de chiffrement en transit n'est pas immédiat et peut prendre un certain temps. Cela est particulièrement vrai pour les grands clusters. Ce n'est que lorsque le cluster termine la migration vers TLS préféré qu'il est en mesure d'accepter et de gérer à la fois les connexions TCP et TLS. Par conséquent, vous ne devez pas créer de clients qui essaient d'établir des connexions TLS avec le cluster tant que le chiffrement en transit n'est pas terminé.

Il existe plusieurs manières d'être averti lorsque le chiffrement en transit est terminé avec succès ou a échoué : (non illustré dans l'exemple de code ci-dessus) :

- Utilisation du service SNS pour recevoir une notification lorsque le chiffrement est terminé
- Utilisation de l'API `describe-events` qui émet un événement lorsque le chiffrement est terminé
- Affichage d'un message dans la ElastiCache console indiquant que le chiffrement est terminé

Vous pouvez également implémenter une logique dans votre application pour savoir si le chiffrement est terminé. Dans l'exemple ci-dessus, nous avons vu plusieurs manières de nous assurer que le cluster termine la migration :

- Attendre le début du processus de migration (le statut du cluster passe à « en cours de modification ») et attendre que la modification soit terminée (le statut du cluster redevient « disponible »)
- Affirmer que le cluster a `transit_encryption_enabled` défini sur True en interrogeant l'API `describe-replication-group`.

Après avoir activé le chiffrement en transit : veillez à ce que les clients que vous utilisez soient correctement configurés

Lorsque le cluster est en mode TLS préféré, votre application doit ouvrir des connexions TLS vers le cluster et utiliser uniquement ces connexions. Ainsi, votre application ne connaît pas d'interruption lors de l'activation du chiffrement en transit. Vous pouvez vous assurer qu'aucune connexion TCP n'est plus claire avec le moteur Valkey ou Redis OSS à l'aide de la commande info dans la section SSL.

```
# SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0   (should be zero)
tls_mode_connected_tls_clients:100
```

Connexion à des nœuds activés avec le chiffrement en transit à l'aide d'OpenSSL (Memcached)

Pour accéder aux données depuis ElastiCache les nœuds Memcached dotés du chiffrement en transit, vous devez utiliser des clients qui fonctionnent avec le protocole SSL (Secure Socket Layer). Vous pouvez également utiliser openssl s_client sur Amazon Linux et Amazon Linux 2.

Pour utiliser openssl s_client afin de se connecter à un cluster Memcached activé avec le chiffrement en transit sur Amazon Linux 2 ou Amazon Linux :

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Création d'un client Memcached en mode TLS avec Java

Pour créer un client en mode TLS, procédez comme suit pour initialiser le client avec le code approprié : SSLContext

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
    public static void main(String[] args) throws Exception {
```

```
        ConnectionFactoryBuilder connectionFactoryBuilder = new
 ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
 TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
 AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

Création d'un client Memcached en mode TLS avec PHP

Pour créer un client en mode TLS, procédez comme suit pour initialiser le client avec le code approprié : SSLContext

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Descovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/
AmazonElastiCache/latest/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example     */
$server_port = 11211;
```

```
/* Initialize a persistent Memcached client and configure it with the Dynamic client
 mode  */
$tls_client =  new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
 it to your client.
 * Note:  These TLS context configurations will be applied to all the servers connected
 to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

 /* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
```

```
    exit(1);
}
```

Pour plus d'informations sur l'utilisation du client PHP, consultez Installation du client de ElastiCache cluster pour PHP.

Chiffrement au repos dans ElastiCache

Pour garantir la sécurité de vos données, Amazon ElastiCache et Amazon S3 proposent différentes méthodes pour restreindre l'accès aux données de votre cache. Pour plus d'informations, consultez Amazon VPCs et la ElastiCache sécurité et Identity and Access Management pour Amazon ElastiCache.

ElastiCache le chiffrement au repos est une fonctionnalité qui permet de renforcer la sécurité des données en chiffrant les données sur disque. Il est toujours activé sur un cache sans serveur. Lorsque cette fonctionnalité est activée, elle chiffre les aspects suivants :

- Le disque lors des opérations de synchronisation, de sauvegarde et d'échange
- Sauvegardes stockées dans Amazon S3

Les données stockées sur SSDs (disques SSD) dans des clusters compatibles avec la hiérarchisation des données sont toujours chiffrées.

ElastiCache propose un chiffrement par défaut (géré par le service) au repos, ainsi que la possibilité d'utiliser vos propres clés AWS KMS symétriques gérées par le client dans le service de gestion des AWS clés (KMS). Lorsque le cache est sauvegardé, sous les options de chiffrement, choisissez d'utiliser la clé de chiffrement par défaut ou une clé gérée par le client. Pour de plus amples informations, veuillez consulter Activation du chiffrement au repos.

> **Important**
>
> L'activation du chiffrement au repos sur un cluster Valkey ou Redis OSS autoconçu implique la suppression de votre groupe de réplication existant, après avoir exécuté la sauvegarde et la restauration sur le groupe de réplication.

Le chiffrement au repos ne peut être activé sur un cache que lorsqu'il est créé. Puisque du traitement est nécessaire pour chiffrer et déchiffrer les données, activer le chiffrement au repos peut avoir un certain impact sur les performances durant ces opérations. Vous devez référencer vos données avec et sans le chiffrement au repos pour déterminer l'impact sur la performance pour vos cas d'utilisation.

Rubriques

- Conditions du chiffrement au repos
- Utilisation de clés gérées par le client à partir de AWS KMS

- Activation du chiffrement au repos
- consultez aussi

Conditions du chiffrement au repos

Les contraintes suivantes relatives ElastiCache au chiffrement au repos doivent être prises en compte lorsque vous planifiez la mise en œuvre du ElastiCache chiffrement au repos :

- Le chiffrement au repos est pris en charge sur les groupes de réplication exécutant Valkey 7.2 et versions ultérieures, ainsi que sur les versions Redis OSS (version 3.2.6 prévue pour EOL, voir calendrier de fin de vie des versions Redis OSS), 4.0.10 ou versions ultérieures.
- Le chiffrement au repos est pris en charge uniquement pour les groupes de réplication s'exécutant dans un Amazon VPC.
- Le chiffrement au repos est pris en charge uniquement pour les groupes de réplication exécutant les types de nœud suivants.
  - R7g, R6gd, R6g, R5, R4, R3
  - M7g, M6g, M5, M4, M3
  - T4, T3, T2
  - C7gn

  Pour plus d'informations, consultez Types de nœuds pris en charge
- Le chiffrement au repos est activé par la définition explicite du paramètre `AtRestEncryptionEnabled` sur `true`.
- Le chiffrement au repos peut être activé sur un groupe de réplication uniquement lorsque vous créez le groupe de réplication. Vous ne pouvez pas activer et désactiver le chiffrement au repos en modifiant un groupe de réplication. Pour plus d'informations sur l'implémentation du chiffrement au repos sur un groupe de réplication existant, consultez Activation du chiffrement au repos.
- Si un cluster utilise un type de nœud de la famille r6gd, les données stockées sur SSD sont chiffrées, que le chiffrement au repos soit activé ou non.
- L'option permettant d'utiliser une clé gérée par le client pour le chiffrement au repos n'est pas disponible dans les régions AWS GovCloud (us-gov-east us-gov-west-1 et -1).
- Si un cluster utilise un type de nœud de la famille r6gd, les données stockées sur un SSD sont chiffrées à l'aide de la clé AWS KMS gérée par le client choisie (ou du chiffrement géré par le service dans les régions). AWS GovCloud

- Avec Memcached, le chiffrement au repos n'est pris en charge que sur les caches sans serveur.
- Lorsque vous utilisez Memcached, l'option permettant d'utiliser une clé gérée par le client pour le chiffrement au repos n'est pas disponible dans les AWS GovCloud régions (us-gov-east-1 et us-gov-west -1).

L'implémentation du chiffrement au repos peut réduire la performance lors des opérations de sauvegarde et de synchronisation de nœud. Référencez le chiffrement au repos en comparaison avec l'absence de chiffrement sur vos propres données pour déterminer l'impact sur la performance pour votre cas d'utilisation.

Utilisation de clés gérées par le client à partir de AWS KMS

ElastiCache prend en charge les clés AWS KMS 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 de plus amples informations, veuillez consulter Clés AWS KMS dans le Guide du développeur AWS Key Management Service. Les clés doivent être créées dans AWS KMS avant de pouvoir être utilisées avec ElastiCache.

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.

ElastiCache vous permet de vous intégrer à AWS KMS. 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 ElastiCache l'intégration d'Amazon à AWS KMS.

La clé de `kms:ViaService` condition limite l'utilisation d'une clé AWS KMS (clé KMS) aux demandes provenant de AWS services spécifiques. À utiliser `kms:ViaService` avec ElastiCache, incluez les deux ViaService noms dans la valeur de la clé de condition : `elasticache.AWS_region.amazonaws.com` et`dax.AWS_region.amazonaws.com`. Pour plus d'informations, voir kms : ViaService.

Vous pouvez l'AWS CloudTrailutiliser pour suivre les demandes qu'Amazon ElastiCache envoie AWS Key Management Service en votre 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 ElastiCache créées en appelant l'appel d'API ListGrantsKMS.

Une fois un groupe de réplication chiffré à l'aide d'une clé gérée par le client, toutes les sauvegardes pour le groupe de réplication sont chiffrées comme suit :

- Les sauvegardes quotidiennes automatiques sont chiffrées à l'aide de la clé gérée par le client associée au cluster.
- La sauvegarde finale créée quand le groupe de réplication est supprimé est également chiffrée à l'aide de la clé gérée par le client associée au groupe de réplication.
- Les sauvegardes manuellement créées sont chiffrées par défaut pour utiliser la clé KMS associée au groupe de réplication. Vous pouvez la remplacer en choisissant une autre clé gérée par le client.
- Copier une sauvegarde revient à utiliser une clé gérée par le client associée à la sauvegarde de la 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 lorsque vous exportez des sauvegardes vers le compartiment Amazon S3 sélectionné. Cependant, toutes les sauvegardes exportées vers Amazon S3 sont chiffrées à l'aide du Chiffrement côté client. Vous pouvez choisir de copier le fichier de sauvegarde vers un nouvel objet S3 et le chiffrer à l'aide d'une clé KMS gérée par le client, copier le fichier vers un autre compartiment S3 configuré avec le chiffrement par défaut à l'aide d'une clé KMS ou changer une option de chiffrement dans le fichier lui-même.
> - Vous pouvez toujours utiliser des clés gérées par le client pour chiffrer des sauvegardes manuellement créées pour les groupes de réplication qui n'utilisent pas de clés gérées par le client pour le chiffrement. Avec cette option, le fichier de sauvegarde stocké dans Amazon S3 est chiffré à l'aide d'une clé KMS, même quand les données ne sont pas chiffrées sur le groupe de réplication original.
>
> La restauration à partir d'une sauvegarde vous permet de choisir parmi les options de chiffrement disponibles, comme pour les choix de chiffrement disponibles lors de la création d'un nouveau groupe de réplication.

- Si vous supprimez la clé ou si vous la désactivez et révoquez les octrois pour la clé que vous avez utilisée pour chiffrer un cache, ce dernier 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 afin de créer une sauvegarde à des fins d'archive.

- 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 ElastiCache données. Les ElastiCache caches Amazon 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 Rotation des clés AWS KMS dans le Guide du développeur du service de gestion des AWS clés.
- Le chiffrement d'un ElastiCache cache à l'aide d'une clé KMS nécessite une autorisation par cache. Cet octroi est utilisé pendant toute la durée de vie du cache. De plus, un octroi par sauvegarde est utilisé pendant la création de la sauvegarde. Cet octroi est retiré une fois la sauvegarde créée.
- Pour plus d'informations sur les AWS autorisations et les limites KMS, consultez la section Limites du guide du développeur du service de gestion des AWS clés.

Activation du chiffrement au repos

Le chiffrement au repos est activé pour tous les caches sans serveur.

Lorsque vous créez un cluster auto-conçu, vous pouvez activer le chiffrement au repos en définissant le paramètre `AtRestEncryptionEnabled` sur `true`. Vous ne pouvez pas activer le chiffrement au repos sur les groupes de réplication existants.

Vous pouvez activer le chiffrement au repos lorsque vous créez un ElastiCache cache. Vous pouvez le faire à l'aide de AWS Management Console, de AWS CLI, ou de l' ElastiCache API.

Lorsque vous créez un cache, vous pouvez sélectionner l'une des options suivantes :

- Par défaut : cette option utilise le chiffrement au repos géré par service.
- Clé gérée par le client : cette option vous permet de fournir la clé ID/ARN depuis AWS KMS pour le chiffrement au repos.

Pour savoir comment créer des clés racines AWS KMS, consultez la section Créer des clés dans le guide du développeur du service de gestion des AWS clés

Table des matières

- Activation du chiffrement au repos à l'aide du AWS Management Console
- Activation du chiffrement au repos à l'aide du AWS CLI

Activation du chiffrement au repos sur un cluster Valkey ou Redis OSS autoconçu existant

Vous ne pouvez activer le chiffrement au repos que lorsque vous créez un groupe de réplication Valkey ou Redis OSS. Si vous disposez déjà d'un groupe de réplication sur lequel vous souhaitez activer le chiffrement au repos, procédez comme suit.

Pour activer le chiffrement au repos sur un groupe de réplication existant

1. Créez une sauvegarde manuelle de votre groupe de réplication existant. Pour de plus amples informations, veuillez consulter Réalisation de sauvegardes manuelles.
2. Créez un groupe de réplication en restaurant la sauvegarde. Sur le nouveau groupe de réplication, activez le chiffrement au repos. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.
3. Dans votre application, mettez à jour les points de terminaison pour pointer vers le nouveau groupe de réplication.
4. Supprimez l'ancien groupe de réplication. Pour plus d'informations, consultez Supprimer un cluster dans ElastiCache ou Suppression d'un groupe de réplication.

Activation du chiffrement au repos à l'aide du AWS Management Console

Activation du chiffrement au repos sur un cache sans serveur (console)

Le chiffrement au repos est activé pour tous les caches sans serveur. Par défaut, une clé KMS AWS appartenant à l'utilisateur est utilisée pour chiffrer les données. Pour choisir votre propre AWS KMS clé, effectuez les sélections suivantes :

- Développez la section Paramètres par défaut.
- Choisissez Personnaliser les paramètres par défaut dans la section Paramètres par défaut.
- Choisissez Personnaliser vos paramètres de sécurité dans la section Sécurité.
- Choisissez Clé CMK gérée par le client sous le paramètre Clé de chiffrement.
- Sélectionnez une clé sous le paramètre CléAWS KMS .

Activation du chiffrement au repos sur un cluster auto-conçu (console)

Lorsque vous concevez votre propre cache, le chiffrement au repos est activé à l'aide de la clé Par défaut pour les configurations « Dev/Test » et « Production » avec la méthode « Création facile ». Lorsque vous choisissez vous-même la configuration, effectuez les sélections suivantes :

- Choisissez la version 3.2.6, 4.0.10 ou ultérieure en tant que version de moteur.
- Cochez la case en regard de Activer pour l'option Chiffrement au repos.
- Choisissez Clé par défaut ou Clé CMK gérée par le client.

Pour la step-by-step procédure, reportez-vous à la section suivante :

- Création d'un cluster Valkey (mode cluster désactivé) (console)
- Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Activation du chiffrement au repos à l'aide du AWS CLI

Pour activer le chiffrement au repos lors de la création d'un cluster Valkey ou Redis OSS à l'aide du AWS CLI at-rest-encryption-enabled paramètre -- lors de la création d'un groupe de réplication.

Activation du chiffrement au repos sur un cluster (CLI) Valkey ou Redis OSS (mode cluster désactivé)

L'opération suivante crée le groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) `my-classic-rg` avec trois nœuds (-- num-cache-clusters), une réplique principale et deux répliques en lecture. Le chiffrement au repos est activé pour ce groupe de réplication (-- at-rest-encryption-enabled).

Les paramètres suivants, ainsi que leurs valeurs, sont nécessaires à l'activation du chiffrement sur ce groupe de réplication :

Paramètres clés

- **`--engine`**—Ça doit être `valkey` ou. `redis`
- **`--engine-version`**—Si le moteur est Redis OSS, il doit être 3.2.6, 4.0.10 ou version ultérieure.
- **`--at-rest-encryption-enabled`** : requis pour activer le chiffrement au repos.

Example 1 : Cluster Valkey ou Redis OSS (mode cluster désactivé) avec répliques

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
    --replication-group-id my-classic-rg \
    --replication-group-description "3 node replication group" \
```

```
    --cache-node-type cache.m4.large \
    --engine redis \
    --at-rest-encryption-enabled \
    --num-cache-clusters 3
```

Pour Windows :

```
aws elasticache create-replication-group ^
    --replication-group-id my-classic-rg ^
    --replication-group-description "3 node replication group" ^
    --cache-node-type cache.m4.large ^
    --engine redis ^
    --at-rest-encryption-enabled ^
    --num-cache-clusters 3 ^
```

Pour plus d'informations, consultez les éléments suivants :

- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro ()AWS CLI
- create-replication-group

Activation du chiffrement au repos sur un cluster pour Valkey ou Redis OSS (mode cluster activé) (CLI)

L'opération suivante crée le groupe de réplication Valkey ou Redis OSS (mode cluster activé) `my-clustered-rg` avec trois groupes de nœuds ou partitions (--). num-node-groups Chacun possède trois nœuds, un nœud principal et deux répliques de lecture (-- replicas-per-node-group). Le chiffrement au repos est activé pour ce groupe de réplication (-- at-rest-encryption-enabled).

Les paramètres suivants, ainsi que leurs valeurs, sont nécessaires à l'activation du chiffrement sur ce groupe de réplication :

Paramètres clés

- **`--engine`**—Ça doit être `valkey` ou. `redis`
- **`--engine-version`**—Si le moteur est Redis OSS, il doit être 4.0.10 ou version ultérieure.
- **`--at-rest-encryption-enabled`** : requis pour activer le chiffrement au repos.

- **`--cache-parameter-group`** : doit indiquer `default-redis4.0.cluster.on` ou l'un de ses dérivés afin de faire de ce cluster un groupe de réplication activé sur le mode.

Example 2 : Un cluster Valkey ou Redis OSS (mode cluster activé)

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
   --replication-group-id my-clustered-rg \
   --replication-group-description "redis clustered cluster" \
   --cache-node-type cache.m3.large \
   --num-node-groups 3 \
   --replicas-per-node-group 2 \
   --engine redis \
   --engine-version 6.2 \
   --at-rest-encryption-enabled \
   --cache-parameter-group default.redis6.x.cluster.on
```

Pour Windows :

```
aws elasticache create-replication-group ^
   --replication-group-id my-clustered-rg ^
   --replication-group-description "redis clustered cluster" ^
   --cache-node-type cache.m3.large ^
   --num-node-groups 3 ^
   --replicas-per-node-group 2 ^
   --engine redis ^
   --engine-version 6.2 ^
   --at-rest-encryption-enabled ^
   --cache-parameter-group default.redis6.x.cluster.on
```

Pour plus d'informations, consultez les éléments suivants :

- Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster activé) à partir de zéro ()AWS CLI
- create-replication-group

consultez aussi

- Amazon VPCs et la ElastiCache sécurité

- Identity and Access Management pour Amazon ElastiCache

Authentification et autorisation

AWS Identity and Access Management (IAM) est un service Web qui vous permet de contrôler en toute sécurité l'accès aux ressources. AWS ElastiCache prend en charge l'authentification des utilisateurs à l'aide d'IAM et de la commande Valkey et Redis OSS AUTH, et l'autorisation des opérations des utilisateurs à l'aide du contrôle d'accès basé sur les rôles (RBAC).

Rubriques

- Contrôle d'accès basé sur les rôles (RBAC)
- Authentification avec la commande Valkey et Redis OSS AUTH
- Désactivation du contrôle d'accès sur un cache ElastiCache Valkey ou Redis OSS

Contrôle d'accès basé sur les rôles (RBAC)

Avec la commande Valkey et Redis OSS AUTH décrite dans la section, Authentification avec la commande Valkey et Redis OSS AUTH vous pouvez utiliser le contrôle d'accès basé sur les rôles (RBAC). Le RBAC est également le seul moyen de contrôler l'accès aux caches sans serveur. Ceci est disponible pour Valkey 7.2 et versions ultérieures, ainsi que pour Redis OSS 6.0 à 7.2.

Le RBAC vous permet de :

- Contrôlez l'accès au cache par le biais de groupes d'utilisateurs. Ces groupes d'utilisateurs sont conçus comme un moyen d'organiser l'accès aux caches.
- Avec AuthN, utilisez des mots de passe par utilisateur plutôt que par des jetons d'authentification par cluster.
- Avec AuthZ, affinez les autorisations utilisateur.
- Basez l'accès à votre cluster sur ACLs.

Contrairement à Valkey et Redis OSS AUTH, où tous les clients authentifiés ont un accès complet au cache si leur jeton est authentifié, le RBAC vous permet d'affecter des utilisateurs à des ensembles en fonction des rôles souhaités par les utilisateurs. Ces ensembles sont conçus pour organiser l'accès aux caches.

Avec RBAC, vous créez des utilisateurs et leur attribuez des autorisations spécifiques à l'aide d'une chaîne d'accès, comme décrit ci-dessous. Vous assignez les utilisateurs à des ensembles

correspondant à un rôle spécifique (administrateurs, ressources humaines) qui sont ensuite déployés dans un ou plusieurs ElastiCache caches. Ce faisant, vous pouvez établir des limites de sécurité entre les clients utilisant le ou les mêmes caches Valkey ou Redis OSS, et empêcher les clients d'accéder aux données des autres.

Le RBAC est conçu pour prendre en charge l'introduction de l'ACL dans Redis OSS 6. Lorsque vous utilisez RBAC avec votre cache ElastiCache Valkey ou Redis OSS, certaines limites s'appliquent :

- Un groupe d'utilisateurs configuré pour le moteur « VALKEY » ne peut contenir que des utilisateurs utilisant un mécanisme d'authentification (mot de passe ou IAM). Cela signifie que tous les utilisateurs du moteur « VALKEY », ainsi que tous les autres utilisateurs du moteur « Redis » dont la configuration est configurée pour s'authentifier par mot de passe ou IAM, peuvent faire partie de ce groupe d'utilisateurs.
- Lors de l'utilisation de RBAC avec des clusters Valkey, les deux groupes d'utilisateurs avec le moteur « VALKEY » et avec le moteur « REDIS » peuvent être utilisés.
- Lorsque vous utilisez RBAC avec des clusters Redis OSS, seuls les groupes d'utilisateurs dotés du moteur « REDIS » peuvent être utilisés.
- Vous ne pouvez pas spécifier de mots de passe dans une chaîne d'accès. Vous définissez des mots de passe avec CreateUserou par ModifyUserappels.
- Pour les droits des utilisateurs, vous activez ou désactivez les utilisateurs à l'aide du on et `off` dans le cadre de la chaîne d'accès. Si ni l'un ni l'autre n'est spécifié dans la chaîne d'accès, l'utilisateur se voit attribuer `off` et ne dispose pas de droits d'accès au cache.
- Vous ne pouvez pas utiliser de commandes interdites ou renommées dans le cadre de la chaîne d'accès. Si vous spécifiez une commande interdite ou renommée, une exception est émise. Si vous souhaitez utiliser des listes de contrôle d'accès (ACLs) pour une commande renommée, spécifiez le nom d'origine de la commande, c'est-à-dire le nom de la commande avant qu'elle ne soit renommée.
- 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 ElastiCache pour Valkey et Redis, OSS 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 ACL appliquées à chaque utilisateur. ElastiCache prend en charge la `ACL LIST` commande, mais n'inclut pas le support pour le hachage des mots de passe comme le fait Redis OSS. Avec ElastiCache, 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 ElastiCache Valkey et Redis OSS incluent ACL WHOAMI, ACL USERS et ACL CAT. ElastiCache pour Valkey et Redis OSS ne prend en charge aucune autre commande ACL basée sur l'écriture.
- Les limites suivantes s'appliquent :

Ressource	Maximum autorisé
Utilisateurs par groupe d'utilisateurs	100
Nombre d'utilisateurs	1 000
Nombre de groupes d'utilisateurs	100

RBAC avec Valkey

Lorsque vous utilisez le contrôle d'accès basé sur les rôles avec Valkey, les utilisateurs et les groupes d'utilisateurs sont créés avec le type de moteur « VALKEY ». Cela est recommandé, car par défaut, Valkey avec RBAC offre une sécurité accrue par rapport à Redis OSS. Les clusters Valkey provisionnés et sans serveur prennent en charge les associations d'utilisateurs et de groupes d'utilisateurs VALKEY.

Les principales fonctionnalités de Valkey Access Control incluent :

- Les utilisateurs de Valkey sont limités aux associations de groupes d'utilisateurs Valkey uniquement.
- Les groupes d'utilisateurs Valkey peuvent contenir des utilisateurs Valkey et des utilisateurs Redis OSS protégés par mot de passe ou activés pour l'authentification IAM.
- Les utilisateurs de Valkey doivent utiliser la protection par mot de passe ou l'authentification IAM.
- Les groupes d'utilisateurs VALKEY ne peuvent être associés qu'aux clusters de cache VALKEY
- Il n'y a aucune exigence utilisateur par défaut. Lorsque le groupe d'utilisateurs Valkey est attaché à des clusters de cache, l'exigence utilisateur par défaut est automatiquement désactivée. Les clients verront que l'utilisateur par défaut est désactivé lorsqu'ils utilisent la commande ACL LIST.

Vous trouverez ci-dessous de plus amples informations sur l'utilisation de RBAC ElastiCache pour Valkey et Redis OSS.

Rubriques

- Définition des autorisations à l'aide d'une chaîne d'accès
- Appliquer le RBAC à un cache ElastiCache pour Valkey ou Redis OSS
- Migration d'AUTH vers RBAC
- Migration de RBAC vers AUTH
- Rotation automatique des mots de passe pour les utilisateurs
- Authentification avec IAM

Définition des autorisations à l'aide d'une chaîne d'accès

Pour spécifier les autorisations d'accès à un cache ElastiCache Valkey ou Redis OSS, vous créez une chaîne d'accès et vous l'attribuez à un utilisateur via 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 s'il y a des redondances 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.
- `+@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 groupes d'utilisateurs avec la console et la CLI.

Si vous migrez une charge de travail existante vers ElastiCache, vous pouvez récupérer la chaîne d'accès en appelant`ACL LIST`, en excluant l'utilisateur et tout hachage de mot de passe.

Pour les versions 6.2 et supérieures de Redis OSS, la syntaxe de chaîne d'accès suivante est également prise en charge :

- `&*` : l'accès est accordé à tous les canaux disponibles.

Pour les versions 7.0 et supérieures de Redis OSS, la syntaxe de chaîne d'accès suivante est également prise en charge :

- `|` : peut être utilisé pour bloquer des sous-commandes (par exemple « -config|set »).
- `%R~<pattern>` : ajoute le modèle de clé de lecture spécifié. Il se comporte de la même manière que le modèle de clé normal, mais il n'autorise que la lecture à partir de clés correspondant au modèle donné. Pour plus d'informations, consultez key permissions (Autorisations de clés).
- `%W~<pattern>` : ajoute le modèle de clé d'écriture spécifié. Il se comporte de la même manière que le modèle de clé normal, mais il n'autorise que l'écriture à partir de clés correspondant au modèle donné. Consultez la section Autorisations relatives aux clés ACL pour plus d'informations.
- `%RW~<pattern>` : alias pour `~<pattern>`.
- `(<rule list>)` : crée un nouveau sélecteur auquel comparer les règles. Les sélecteurs sont évalués en fonction des autorisations utilisateur et de l'ordre dans lequel ils sont définis. Si une commande correspond aux autorisations utilisateur ou à n'importe quel sélecteur, elle est autorisée. Pour plus d'informations, consultez ACL selectors (Sélecteurs ACL).
- `clearselectors` : supprime tous les sélecteurs associés à l'utilisateur.

Appliquer le RBAC à un cache ElastiCache pour Valkey ou Redis OSS

ElastiCache Pour l'utiliser pour Valkey ou Redis OSS RBAC, vous devez suivre les étapes suivantes :

1. Créez un ou plusieurs utilisateurs.
2. Créez un groupe d'utilisateurs puis ajoutez-y des utilisateurs.
3. Attribuez le groupe d'utilisateurs à un cache ayant le chiffrement en transit activé.

Ces étapes sont décrites en détail comme suit.

Rubriques

- Création d'utilisateurs et de groupes d'utilisateurs avec la console et la CLI
- Gestion des groupes d'utilisateurs avec la console et la CLI
- Attribution de groupes d'utilisateurs à des caches sans serveur
- Attribution de groupes d'utilisateurs à des groupes de réplication

Création d'utilisateurs et de groupes d'utilisateurs avec la console et la CLI

Les informations utilisateur pour les utilisateurs RBAC sont un ID utilisateur, 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. L'ID utilisateur est unique à l'utilisateur, et le nom d'utilisateur est celui qui est transmis au moteur.

Assurez-vous que les autorisations d'utilisateur que vous fournissez correspondent à l'objectif visé du groupe d'utilisateurs. Par exemple, si vous créez un groupe d'utilisateurs appelé `Administrators`, tout utilisateur que vous ajoutez à ce groupe doit avoir sa chaîne d'accès définie pour un accès complet aux clés et aux commandes. Pour les utilisateurs d'un groupe d'utilisateurs `e-commerce`, vous pouvez définir leurs chaînes d'accès en lecture seule.

ElastiCache configure automatiquement un utilisateur par défaut avec son ID utilisateur et son nom d'utilisateur `"default"` et l'ajoute à tous les groupes d'utilisateurs. 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.

Pour ajouter un contrôle d'accès approprié à un cache, remplacez cet utilisateur par défaut par un nouvel utilisateur qui n'est pas activé ou qui utilise un mot de passe fort. Pour modifier l'utilisateur par

défaut, créez un nouvel utilisateur avec le nom d'utilisateur défini sur `default`. Vous pouvez ensuite l'échanger avec l'utilisateur par défaut d'origine.

Les procédures suivantes montrent comment échanger l'utilisateur `default` d'origine avec un autre utilisateur `default` qui a une chaîne d'accès modifiée.

Pour modifier l'utilisateur par défaut sur la console

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Choisissez Gestion des groupes d'utilisateurs dans le volet de navigation.
3. Pour ID de groupe d'utilisateurs, choisissez l'ID que vous souhaitez modifier. Assurez-vous de sélectionner le lien et non pas la case à cocher.
4. Sélectionnez Modifier.
5. Dans la fenêtre Modifier, choisissez Gérer. Pour « sélectionner l'utilisateur que vous souhaitez », sélectionnez l'utilisateur dont le nom d'utilisateur est le nom d'utilisateur par défaut.
6. Choisissez Choisir.
7. Sélectionnez Modifier. Lorsque vous effectuez cette opération, toutes les connexions existantes à un cache que possède l'utilisateur par défaut d'origine sont interrompues.

Pour modifier l'utilisateur par défaut à l'aide du AWS CLI

1. Créez un utilisateur avec le nom d'utilisateur `default` à l'aide des commandes suivantes.

   Pour Linux, macOS ou Unix :

   ```
   aws elasticache create-user \
    --user-id "new-default-user" \
    --user-name "default" \
    --engine "VALKEY" \
    --passwords "a-str0ng-pa))word" \
    --access-string "off +get ~keys*"
   ```

   Pour Windows :

   ```
   aws elasticache create-user ^
    --user-id "new-default-user" ^
    --user-name "default" ^
   ```

```
 --engine "VALKEY" ^
 --passwords "a-str0ng-pa))word" ^
 --access-string "off +get ~keys*"
```

2. Créez un groupe d'utilisateurs et ajoutez l'utilisateur que vous avez créé précédemment.

   Pour Linux, macOS ou Unix :

```
aws elasticache create-user-group \
  --user-group-id "new-group-2" \
  --engine "VALKEY" \
  --user-ids "new-default-user"
```

   Pour Windows :

```
aws elasticache create-user-group ^
  --user-group-id "new-group-2" ^
  --engine "VALKEY" ^
  --user-ids "new-default-user"
```

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 caches sont maintenues.

Tenez compte en particulier de ces contraintes liées au mot de passe utilisateur lorsque vous utilisez RBAC ElastiCache pour Valkey et Redis OSS :

- 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

Utilisez la procédure suivante pour gérer les utilisateurs sur la console.

Pour gérer les utilisateurs sur la console

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Sur le ElastiCache tableau de bord Amazon, choisissez Gestion des utilisateurs. Les options suivantes sont disponibles :

- Créer un utilisateur : lors de la création d'un utilisateur, vous entrez un ID utilisateur, un nom d'utilisateur, un mode d'authentification et une chaîne d'accès. La chaîne d'accès définit le niveau d'autorisation accordé à l'utilisateur pour les clés et commandes.

  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 caches sont maintenues.
- Modifier l'utilisateur : permet de mettre à jour les paramètres d'authentification d'un utilisateur ou de modifier sa chaîne d'accès.
- Supprimer un utilisateur : le compte sera supprimé de tous les groupes d'utilisateurs auxquels il appartient.

Utilisez les procédures suivantes pour gérer les utilisateurs avec l' AWS CLI.

Pour modifier un utilisateur à l'aide de la CLI

- Utilisez la commande `modify-user` pour mettre à jour le mot de passe ou les mots de passe d'un utilisateur ou modifier les autorisations d'accès d'un utilisateur.

  Lorsqu'un utilisateur est modifié, les groupes d'utilisateurs associés à l'utilisateur sont mis à jour, ainsi que les caches associés au groupe d'utilisateurs. Toutes les connexions existantes sont maintenues. Voici quelques exemples.

  Pour Linux, macOS ou Unix :

  ```
  aws elasticache modify-user \
    --user-id user-id-1 \
    --access-string "~objects:* ~items:* ~public:*" \
    --authentication-mode Type=iam
  ```

  Pour Windows :

  ```
  aws elasticache modify-user ^
    --user-id user-id-1 ^
    --access-string "~objects:* ~items:* ~public:*" ^
    --authentication-mode Type=iam
  ```

> **Note**
>
> Nous vous déconseillons d'utiliser l'option `nopass`. Si vous le faites, nous vous recommandons de définir les autorisations de l'utilisateur en lecture seule avec accès à un ensemble limité de clés.

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 des groupes d'utilisateurs auxquels il appartient. Voici un exemple.

  Pour Linux, macOS ou Unix :

  ```
  aws elasticache delete-user \
    --user-id user-id-2
  ```

  Pour Windows :

  ```
  aws elasticache delete-user ^
    --user-id user-id-2
  ```

Pour afficher une liste des utilisateurs, appelez l'opération describe-users.

```
aws elasticache describe-users
```

Gestion des groupes d'utilisateurs avec la console et la CLI

Vous pouvez créer des groupes d'utilisateurs pour organiser et contrôler l'accès des utilisateurs à un ou plusieurs caches, comme indiqué ci-dessous.

Utilisez la procédure suivante pour gérer des groupes d'utilisateurs à l'aide de la console.

Pour gérer les groupes d'utilisateurs à l'aide de la console

1. Connectez-vous à la ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Sur le ElastiCache tableau de bord Amazon, choisissez User group management.

Les opérations suivantes sont disponibles pour créer de nouveaux groupes d'utilisateurs :

- Créer : lorsque vous créez un groupe d'utilisateurs, vous ajoutez des utilisateurs, puis affectez les groupes d'utilisateurs à des caches. Par exemple, vous pouvez créer un groupe d'utilisateurs `Admin` pour les utilisateurs qui ont des rôles d'administration sur un cache.

  > **Important**
  >
  > Si vous n'utilisez pas de groupe d'utilisateurs Valkey ou Redis OSS, vous devez inclure un utilisateur par défaut lors de la création d'un groupe d'utilisateurs.

- Add Users (Ajouter des utilisateurs) : ajoute des utilisateurs au groupe d'utilisateurs.
- Remove Users (Supprimer des utilisateurs) : supprime des utilisateurs du groupe d'utilisateurs. Lorsque des utilisateurs sont supprimés d'un groupe d'utilisateurs, toutes les connexions existantes qu'ils ont à un cache sont interrompues.
- Delete (Supprimer) : utilisez cette option pour supprimer un groupe d'utilisateurs. Notez que le groupe d'utilisateurs lui-même, et non les utilisateurs appartenant au groupe, sera supprimé.

Pour les groupes d'utilisateurs existants, procédez comme suit :

- Add Users (Ajouter des utilisateurs) : ajoute des utilisateurs existants au groupe d'utilisateurs.
- Delete Users (Supprimer des utilisateurs) : supprime les utilisateurs existants du groupe d'utilisateurs.

  > **Note**
  >
  > Les utilisateurs sont supprimés du groupe d'utilisateurs, mais pas supprimés du système.

Utilisez les procédures suivantes pour gérer des groupes d'utilisateurs à l'aide de la CLI.

Pour créer un groupe d'utilisateurs et ajouter un utilisateur à l'aide de la CLI

- Utilisez la commande `create-user-group`, comme illustré ci-dessous.

  Pour Linux, macOS ou Unix :

```
aws elasticache create-user-group \
  --user-group-id "new-group-1" \
  --engine "VALKEY" \
  --user-ids user-id-1, user-id-2
```

Pour Windows :

```
aws elasticache create-user-group ^
  --user-group-id "new-group-1" ^
  --engine "VALKEY" ^
  --user-ids user-id-1, user-id-2
```

Pour modifier un groupe d'utilisateurs en ajoutant de nouveaux utilisateurs ou en supprimant des membres actuels à l'aide de la CLI

- Utilisez la commande `modify-user-group`, comme illustré ci-dessous.

  Pour Linux, macOS ou Unix :

  ```
  aws elasticache modify-user-group --user-group-id new-group-1 \
  --user-ids-to-add user-id-3 \
  --user-ids-to-remove user-id-2
  ```

  Pour Windows :

  ```
  aws elasticache modify-user-group --user-group-id new-group-1 ^
  --user-ids-to-add user-id-3 ^
  --user-ids-to-removere user-id-2
  ```

> **Note**
>
> Toutes les connexions ouvertes appartenant à un utilisateur supprimé d'un groupe d'utilisateurs sont interrompues par cette commande.

Pour supprimer un groupe d'utilisateurs à l'aide de la CLI

- Utilisez la commande `delete-user-group`, comme illustré ci-dessous. Le groupe d'utilisateurs lui-même, et non les utilisateurs appartenant au groupe, est supprimé.

  Pour Linux, macOS ou Unix :

  ```
  aws elasticache delete-user-group /
     --user-group-id
  ```

  Pour Windows :

  ```
  aws elasticache delete-user-group ^
     --user-group-id
  ```

Pour voir la liste des groupes d'utilisateurs, vous pouvez appeler l'describe-user-groupsopération.

```
aws elasticache describe-user-groups \
  --user-group-id test-group
```

Attribution de groupes d'utilisateurs à des caches sans serveur

Après avoir créé un groupe d'utilisateurs et ajouté des utilisateurs, la dernière étape de la mise en œuvre du RBAC consiste à attribuer le groupe d'utilisateurs à un cache sans serveur.

Attribution de groupes d'utilisateurs à des caches sans serveur à l'aide de la console

Pour ajouter un groupe d'utilisateurs à un cache sans serveur à l'aide du AWS Management Console, procédez comme suit :

- Pour le cluster en mode désactivé, veuillez consulter Création d'un cluster Valkey (mode cluster désactivé) (console)
- Pour le cluster en mode activé, veuillez consulter Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Affectation de groupes d'utilisateurs à des caches sans serveur à l'aide du AWS CLI

L' AWS CLI opération suivante crée un cache sans serveur en utilisant le user-group-id paramètre avec la valeur*my-user-group-id*. Remplacez le groupe de sous-réseaux `sng-test` par un groupe de sous-réseaux existant.

Paramètres clés

- **`--engine`**— Ça doit être `VALKEY` ou`REDIS`.
- **`--user-group-id`** : cette valeur fournit l'ID du groupe d'utilisateurs comprenant les utilisateurs disposant d'autorisations d'accès spécifiées pour le cache.

Pour Linux, macOS ou Unix :

```
aws elasticache create-serverless-cache \
    --serverless-cache-name "new-serverless-cache" \
    --description "new-serverless-cache" \
    --engine "VALKEY" \
    --user-group-id "new-group-1"
```

Pour Windows :

```
aws elasticache create-serverless-cache ^
    --serverless-cache-name "new-serverless-cache" ^
    --description "new-serverless-cache" ^
    --engine "VALKEY" ^
    --user-group-id "new-group-1"
```

L' AWS CLI opération suivante modifie un cache sans serveur avec le user-group-id paramètre associé à la valeur. *my-user-group-id*

Pour Linux, macOS ou Unix :

```
aws elasticache modify-serverless-cache \
    --serverless-cache-name serverless-cache-1 \
    --user-group-id "new-group-2"
```

Pour Windows :

```
aws elasticache modify-serverless-cache ^
```

```
    --serverless-cache-name serverless-cache-1 ^
    --user-group-id "new-group-2"
```

Notez que les modifications apportées à un cache sont mises à jour de manière asynchrone. Vous pouvez surveiller cette progression en consultant les événements. Pour de plus amples informations, veuillez consulter Affichage des ElastiCache événements.

Attribution de groupes d'utilisateurs à des groupes de réplication

Après avoir créé un groupe d'utilisateurs et ajouté des utilisateurs, la dernière étape de la mise en œuvre de RBAC consiste à attribuer le groupe d'utilisateurs à un groupe de réplication.

Attribution de groupes d'utilisateurs à des groupes de réplication à l'aide de la console

Pour ajouter un groupe d'utilisateurs à une réplication à l'aide du AWS Management Console, procédez comme suit :

- Pour le cluster en mode désactivé, veuillez consulter Création d'un cluster Valkey (mode cluster désactivé) (console)
- Pour le cluster en mode activé, veuillez consulter Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console)

Affectation de groupes d'utilisateurs à des groupes de réplication à l'aide du AWS CLI

L' AWS CLI opération suivante crée un groupe de réplication avec le chiffrement en transit (TLS) activé et le user-group-ids paramètre avec la valeur*`my-user-group-id`*. Remplacez le groupe de sous-réseaux `sng-test` par un groupe de sous-réseaux existant.

Paramètres clés

- **`--engine`**— Ça doit être `valkey` ou`redis`.
- **`--engine-version`** : doit être la version 6.0 ou ultérieure.
- **`--transit-encryption-enabled`** : obligatoire pour l'authentification et l'association d'un groupe d'utilisateurs.
- **`--user-group-ids`** : cette valeur fournit l'ID du groupe d'utilisateurs comprenant les utilisateurs disposant d'autorisations d'accès spécifiées pour le cache.
- **`--cache-subnet-group`** : obligatoire pour associer un groupe d'utilisateurs.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
    --replication-group-id "new-replication-group" \
    --replication-group-description "new-replication-group" \
    --engine "VALKEY" \
    --cache-node-type cache.m5.large \
    --transit-encryption-enabled \
    --user-group-ids "new-group-1" \
    --cache-subnet-group "cache-subnet-group"
```

Pour Windows :

```
aws elasticache create-replication-group ^
    --replication-group-id "new-replication-group" ^
    --replication-group-description "new-replication-group" ^
    --engine "VALKEY" ^
    --cache-node-type cache.m5.large ^
    --transit-encryption-enabled ^
    --user-group-ids "new-group-1" ^
    --cache-subnet-group "cache-subnet-group"
```

L' AWS CLI opération suivante modifie un groupe de réplication avec le chiffrement en transit (TLS) activé et le user-group-ids paramètre avec la valeur. *my-user-group-id*

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id replication-group-1 \
    --user-group-ids-to-remove "new-group-1" \
    --user-group-ids-to-add "new-group-2"
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id replication-group-1 ^
    --user-group-ids-to-remove "new-group-1" ^
    --user-group-ids-to-add "new-group-2"
```

Notez le `PendingChanges` dans la réponse. Toutes les modifications apportées à un cache sont mises à jour de manière asynchrone. Vous pouvez surveiller cette progression en consultant les

événements. Pour de plus amples informations, veuillez consulter Affichage des ElastiCache événements.

Migration d'AUTH vers RBAC

Si vous utilisez AUTH comme décrit dans Authentification avec la commande Valkey et Redis OSS AUTH et que vous souhaitez migrer vers l'utilisation de RBAC, suivez les procédures suivantes.

Utilisez la procédure suivante pour migrer d'AUTH vers RBAC à l'aide de la console.

Pour migrer de Valkey ou Redis OSS AUTH vers RBAC à l'aide de la console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste située dans le coin supérieur droit, choisissez la AWS région dans laquelle se trouve le cache que vous souhaitez modifier.
3. Dans le volet de navigation, choisissez le moteur qui s'exécute sur le cache que vous souhaitez modifier.

   La liste des caches pour le moteur choisi s'affiche.
4. Dans la liste des caches, choisissez le nom du cache que vous souhaitez modifier.
5. Pour Actions, choisissez Modifier.

   La fenêtre Modifier apparaît.
6. Pour Contrôle d'accès, choisissez Liste de contrôle d'accès de groupe d'utilisateurs.
7. Pour Liste de contrôle d'accès de groupe d'utilisateurs, choisissez un groupe d'utilisateurs.
8. Choisissez Prévisualiser les modifications, puis Modifier sur l'écran suivant.

Utilisez la procédure suivante pour migrer de Valkey ou Redis OSS AUTH vers RBAC à l'aide de la CLI.

Pour migrer d'AUTH vers RBAC à l'aide de la CLI

- Utilisez la commande `modify-replication-group`, comme illustré ci-dessous.

  Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group --replication-group-id test \
    --auth-token-update-strategy DELETE \
```

```
    --user-group-ids-to-add user-group-1
```

Pour Windows :

```
aws elasticache modify-replication-group --replication-group-id test ^
    --auth-token-update-strategy DELETE ^
    --user-group-ids-to-add user-group-1
```

Migration de RBAC vers AUTH

Si vous utilisez RBAC et que vous souhaitez migrer vers Redis OSS AUTH, consultez. Migration de RBAC vers AUTH

> **Note**
>
> Si vous devez désactiver le contrôle d'accès à un ElastiCache cache, vous devez le faire via le AWS CLI. Pour de plus amples informations, veuillez consulter the section called "Désactivation du contrôle d'accès sur un cache ElastiCache Valkey ou Redis OSS".

Rotation automatique des mots de passe pour les utilisateurs

Vous pouvez AWS Secrets Manager ainsi remplacer les informations d'identification codées en dur dans votre code (y compris les mots de passe) par un appel d'API à Secrets Manager pour récupérer le secret par programmation. Cela permet de garantir que le secret ne peut pas être mis en péril par une personne qui examine votre code, étant donné que le secret n'y figure pas. En outre, vous pouvez configurer Secrets Manager afin d'effectuer automatiquement une rotation du secret, selon une planification que vous spécifiez. Cela vous permet de remplacer les secrets à long terme par ceux à court terme, ce qui réduit considérablement le risque de mise en péril.

À l'aide de Secrets Manager, vous pouvez automatiquement faire pivoter vos ElastiCache mots de passe Redis OSS (c'est-à-dire les secrets) à l'aide d'une AWS Lambda fonction fournie par Secrets Manager.

Pour plus d'informations AWS Secrets Manager, voir Qu'est-ce que c'est AWS Secrets Manager ?

Comment ElastiCache utilise les secrets

Valkey 7.2 et versions ultérieures ont un ensemble de fonctionnalités équivalent à celui de Redis OSS 7.0. Dans Redis OSS 6, ElastiCache introduit Contrôle d'accès basé sur les rôles (RBAC) pour

sécuriser le cluster Valkey ou Redis OSS. Cette fonction permet de limiter certaines connexions en termes de commandes pouvant être exécutées et de clés accessibles. Avec RBAC, lorsque le client crée un utilisateur avec des mots de passe, les valeurs des mots de passe doivent être saisies manuellement en texte clair et visibles de l'opérateur.

Avec Secrets Manager, les applications récupèrent le mot de passe depuis Secrets Manager plutôt que de le saisir manuellement et de le stocker dans la configuration de l'application. Pour plus d'informations sur la procédure à utiliser, consultez Comment ElastiCache les utilisateurs sont associés au secret.

L'utilisation de secrets entraîne des frais. Pour plus d'informations sur la tarification, consultez Tarification d'AWS Secrets Manager.

Comment ElastiCache les utilisateurs sont associés au secret

Secrets Manager conservera une référence pour l'utilisateur associé dans le champ `SecretString` du secret. Il n'y aura aucune référence au secret vu de ElastiCache côté.

```
{
    "password": "strongpassword",
    "username": "user1",
    "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1" //this is the
 bond between the secret and the user
}
```

Fonction de rotation Lambda

Pour activer la rotation automatique des mots de passe de Secrets Manager, vous allez créer une fonction Lambda qui interagira avec l'API modify-user pour mettre à jour les mots de passe de l'utilisateur.

Pour plus d'informations sur la façon dont cela fonctionne, consultez Fonctionnement de la rotation.

> **Note**
>
> Pour certains AWS services, afin d'éviter toute confusion dans le scénario adjoint, il est AWS recommandé d'utiliser à la fois les clés de condition `aws:SourceArn` et les clés de condition `aws:SourceAccount` globale. Cependant, si vous incluez la condition `aws:SourceArn` dans votre stratégie de fonction de rotation Lambda, la fonction de rotation ne peut être utilisée que pour effectuer la rotation du secret spécifié par cet ARN. Nous

vous recommandons d'inclure uniquement la clé de contexte `aws:SourceAccount` afin de pouvoir utiliser la fonction de rotation pour plusieurs secrets.

Pour tout problème que vous pourriez rencontrer, consultez Résoudre les problèmes liés à la rotation de AWS Secrets Manager.

Comment créer un ElastiCache utilisateur et l'associer à Secrets Manager

Les étapes suivantes illustrent comment créer un utilisateur et l'associer à Secrets Manager :

1. Créer un utilisateur inactif

   Pour Linux, macOS ou Unix :

```
aws elasticache create-user \
 --user-id user1 \
 --user-name user1 \
 --engine "REDIS" \
 --no-password \ // no authentication is required
 --access-string "*off* +get ~keys*" // this disables the user
```

   Pour Windows :

```
aws elasticache create-user ^
 --user-id user1 ^
 --user-name user1 ^
 --engine "REDIS" ^
 --no-password ^ // no authentication is required
 --access-string "*off* +get ~keys*" // this disables the user
```

   Vous verrez une réponse similaire à ceci :

```
{
    "UserId": "user1",
    "UserName": "user1",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "off ~keys* -@all +get",
    "UserGroupIds": [],
    "Authentication": {
```

```
        "Type": "no_password"
    },
    "ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
}
```

2. Créer un secret

   Pour Linux, macOS ou Unix :

```
aws secretsmanager create-secret \
--name production/ec/user1 \
--secret-string \
'{
   "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
    "username":"user1"
 }'
```

   Pour Windows :

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
   "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
    "username":"user1"
 }'
```

   Vous verrez une réponse similaire à ceci :

```
{
 "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
 "Name": "production/ec/user1",
 "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

3. Configurer une fonction Lambda pour la rotation de votre mot de passe

   a. Connectez-vous à la console Lambda AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/lambda/

b. Dans le volet de navigation, choisissez Functions (Fonctions), puis la fonction que vous avez créée. Choisissez le nom de la fonction, et non la case à cocher située à gauche.

c. Cliquez sur l'onglet Configuration.

d. Dans General configuration (Configuration générale), choisissez Edit (Modifier), puis définissez Timeout (Expiration) sur 12 minutes minimum.

e. Choisissez Enregistrer.

f. Choisissez Environment variables (Variables d'environnement), puis définissez les paramètres suivants :

   i. SECRETS_MANAGER_ENDPOINT : https://secretsmanager.**REGION**.amazonaws.com

   ii. SECRET_ARN : l'Amazon Resource Name (ARN) du secret que vous avez créé à l'étape 2.

   iii. USER_NAME — Nom d'utilisateur de l' ElastiCache utilisateur,

   iv. Choisissez Enregistrer.

g. Choisissez Permissions (Autorisations).

h. Sous Execution role (Rôle d'exécution), choisissez le nom du rôle de la fonction Lambda à afficher sur la console IAM.

i. La fonction Lambda aura besoin de l'autorisation suivante pour modifier les utilisateurs et définir le mot de passe :

   ElastiCache

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:DescribeUsers",
                "elasticache:ModifyUser"
            ],
            "Resource": "arn:aws:elasticache:us-
east-1:111122223333:user:user1"
        }
    ]
```

```
}
```

Secrets Manager

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue",
                "secretsmanager:UpdateSecretVersionStage"
            ],
            "Resource": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:XXXX"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetRandomPassword",
            "Resource": "*"
        }
    ]
}
```

4. Configurer la rotation des secrets de Secrets Manager

   a. À l'aide de AWS Management Console, voir Configurer la rotation automatique pour les AWS secrets de Secrets Manager à l'aide de la console

      Pour plus d'informations sur comment planifier une rotation, consultez Programmer des expressions en rotation avec Secrets Manager.

   b. À l'aide du AWS CLI, voir Configuration de la rotation automatique pour AWS Secrets Manager utiliser le AWS Command Line Interface

Authentification avec IAM

Rubriques

- Présentation
- Limites
- Configuration
- Connexion

Présentation

Avec l'authentification IAM, vous pouvez authentifier une connexion à ElastiCache Valkey ou Redis OSS à l'aide d'identités AWS IAM, lorsque votre cache est configuré pour utiliser Valkey ou Redis OSS version 7 ou supérieure. Cela vous permet de renforcer votre modèle de sécurité et de simplifier de nombreuses tâches administratives de sécurité. Vous pouvez également utiliser l'authentification IAM pour configurer un contrôle d'accès précis pour chaque ElastiCache cache et chaque ElastiCache utilisateur, conformément aux principes d'autorisation du moindre privilège. L'authentification IAM ElastiCache fonctionne en fournissant un jeton d'authentification IAM de courte durée au lieu d'un mot de passe ElastiCache utilisateur de longue durée dans l'OSS ou la commande Valkey ou 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 Valkey ou Redis OSS. Vous pouvez également accorder l'accès aux utilisateurs depuis leurs fournisseurs d'identité fédérés directement aux caches Valkey ou Redis OSS.

Pour utiliser AWS IAM avec ElastiCache, vous devez d'abord créer un ElastiCache utilisateur dont le mode d'authentification est défini sur IAM. Vous pouvez ensuite créer ou réutiliser une identité IAM. L'identité IAM a besoin d'une politique associée pour accorder l'`elasticache:Connect`action au ElastiCache cache et à l' ElastiCache utilisateur. 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 comme mot de passe dans votre client Valkey ou Redis OSS lorsque vous vous connectez à votre cache. Un client Valkey ou Redis OSS prenant en charge le fournisseur d'informations d'identification peut générer automatiquement les informations d'identification temporaires pour chaque nouvelle connexion. ElastiCache effectuera l'authentification IAM pour les demandes de connexion des ElastiCache utilisateurs 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 ElastiCache de Valkey 7.2 ou version ultérieure ou de Redis OSS version 7.0 et supérieure.
- Pour les ElastiCache utilisateurs compatibles IAM, les propriétés du nom d'utilisateur et de l'identifiant utilisateur doivent être identiques.
- Le jeton d'authentification IAM est valide pendant 15 minutes. Pour les connexions de longue durée, nous vous recommandons d'utiliser un client Valkey ou Redis OSS qui prend en charge une interface de fournisseur d'informations d'identification.
- Une connexion authentifiée IAM ElastiCache pour Valkey ou Redis OSS 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 prend en charge les clés de contexte de condition globale suivantes :
  - Lorsque vous utilisez l'authentification IAM avec des caches sans serveur, `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` et `aws:ResourceTag/%s` (à partir des caches sans serveur et des utilisateurs associés) sont prises en charge.
  - Lorsque vous utilisez l'authentification IAM avec des groupes de réplication, `aws:SourceIp` et `aws:ResourceTag/%s` (à partir des groupes de réplication et des utilisateurs associés) sont prises en charge.

  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. Créez un cache.

```
aws elasticache create-serverless-cache \
  --serverless-cache-name cache-01  \
  --description "ElastiCache IAM auth application" \
```

```
  --engine redis
```

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.

   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.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-
east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

4. Créez un rôle IAM.

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

5. Créez la politique IAM.

```
aws iam create-policy \
  --policy-name "elasticache-allow-all" \
  --policy-document file://policy.json
```

6. Attachez la politique gérée IAM au rôle.

```
aws iam attach-role-policy \
 --role-name "elasticache-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

7. Créez un compte utilisateur prenant en charge IAM.

```
aws elasticache create-user \
  --user-name iam-user-01 \
  --user-id iam-user-01 \
  --authentication-mode Type=iam \
  --engine redis \
  --access-string "on ~* +@all"
```

8. Créez un groupe d'utilisateurs et attachez l'utilisateur.

```
aws elasticache create-user-group \
  --user-group-id iam-user-group-01 \
  --engine redis \
  --user-ids default iam-user-01

aws elasticache modify-serverless-cache \
  --serverless-cache-name cache-01  \
  --user-group-id iam-user-group-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 lorsque vous vous connectez à un cache Valkey ou Redis OSS, comme indiqué dans l'exemple ci-dessous.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
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 ElastiCache
 with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);
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(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.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 PARAM_RESOURCE_TYPE = "ResourceType";
    private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
    private static final String ACTION_NAME = "connect";
    private static final String SERVICE_NAME = "elasticache";
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final String userId;
    private final String cacheName;
    private final String region;
    private final boolean isServerless;

    public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
 isServerless) {
        this.userId = userId;
        this.cacheName = cacheName;
        this.region = region;
        this.isServerless = isServerless;
    }

    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(userId));
        if (isServerless) {
            request.addParameters(PARAM_RESOURCE_TYPE,
 Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
        }
        return request;
    }
```

```
    private URI getRequestUri() {
        return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
    }

    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 à ElastiCache l'aide du fournisseur d'informations d'authentification IAM.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
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 ElastiCache with Redis OSS.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);
```

```
// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
    userId, 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 client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Vous trouverez ci-dessous un exemple de client Lettuce Redis OSS qui intègre un fournisseur d'informations d'identification pour générer automatiquement des informations d'identification temporaires IAMAuth TokenRequest en cas de besoin.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userId;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userId,
        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(userId, iamAuthTokenSupplier.get()));
    }

    private String getIamAuthToken() {
        return
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
}
```

Authentification avec la commande Valkey et Redis OSS AUTH

> **Note**
>
> Le AUTH a été remplacé par. the section called "Contrôle d'accès basé sur les rôles (RBAC)" Tous les caches sans serveur doivent utiliser le contrôle RBAC pour l'authentification.

Les jetons d'authentification ou mots de passe Valkey et Redis OSS permettent à Valkey et Redis OSS d'exiger un mot de passe avant d'autoriser les clients à exécuter des commandes, améliorant ainsi la sécurité des données. Le AUTH n'est disponible que pour les clusters conçus par vos soins.

Rubriques

- Présentation de AUTH ElastiCache pour Valkey et Redis OSS
- Appliquer l'authentification à un cluster ElastiCache pour Valkey et Redis OSS
- Modification du jeton AUTH sur un cluster existant
- Migration de RBAC vers AUTH

Présentation de AUTH ElastiCache pour Valkey et Redis OSS

Lorsque vous utilisez le AUTH avec votre cluster ElastiCache for Valkey et Redis OSS, certaines améliorations sont apportées.

Tenez compte en particulier de ces contraintes liées au jeton AUTH ou au mot de passe lorsque vous utilisez AUTH :

- Les jetons, ou mots de passe, doivent comporter de 16 à 128 caractères imprimables.
- Les caractères non alphanumériques sont limités à (!, &, #, $, ^, <, >, -).

- AUTH ne peut être activé que pour les clusters Valkey ou Redis OSS compatibles avec le chiffrement en transit.

Pour configurer un jeton fort, nous vous recommandons de suivre une stratégie de mot de passe stricte, par exemple en exigeant les éléments suivants :

- Les jetons ou mots de passe doivent inclure au moins trois des types de caractères suivants :
  - Caractères majuscules
  - Caractères minuscules
  - Chiffres
  - Caractères non alphanumériques (`!`, &, #, $, ^, <, >, `-`)
- Les jetons ou mots de passe ne doivent pas contenir un mot du dictionnaire ou un mot du dictionnaire légèrement modifié.
- Les jetons ou mots de passe ne doivent pas être identiques ou similaires à un jeton récemment utilisé.

Appliquer l'authentification à un cluster ElastiCache pour Valkey et Redis OSS

Vous pouvez demander aux utilisateurs de saisir un jeton (mot de passe) sur un serveur Valkey ou Redis OSS protégé par des jetons. Pour ce faire, incluez le paramètre `--auth-token` (API : `AuthToken`) avec le jeton correct lorsque vous créez votre groupe ou cluster de réplication. Incluez-le également dans toutes les commandes suivantes vers le groupe ou le cluster de réplication.

L' AWS CLI opération suivante crée un groupe de réplication avec le chiffrement en transit (TLS) activé et le AUTH jeton*`This-is-a-sample-token`*. Remplacez le groupe de sous-réseaux `sng-test` par un groupe de sous-réseaux existant.

Paramètres clés

- **`--engine`**— Ça doit être `valkey` ou`redis`.
- **`--engine-version`**— Si le moteur est Redis OSS, il doit être 3.2.6, 4.0.10 ou version ultérieure.
- **`--transit-encryption-enabled`** : obligatoire pour l'authentification et l'éligibilité HIPAA.
- **`--auth-token`** : obligatoire pour l'éligibilité HIPAA. Cette valeur doit être le jeton correct pour ce serveur Valkey ou Redis OSS protégé par des jetons.
- **`--cache-subnet-group`** : obligatoire pour l'éligibilité HIPAA.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
    --replication-group-id authtestgroup \
    --replication-group-description authtest \
    --engine redis \
    --cache-node-type cache.m4.large \
    --num-node-groups 1 \
    --replicas-per-node-group 2 \
    --transit-encryption-enabled \
    --auth-token This-is-a-sample-token \
    --cache-subnet-group sng-test
```

Pour Windows :

```
aws elasticache create-replication-group ^
    --replication-group-id authtestgroup ^
    --replication-group-description authtest ^
    --engine redis ^
    --cache-node-type cache.m4.large ^
    --num-node-groups 1 ^
    --replicas-per-node-group 2 ^
    --transit-encryption-enabled ^
    --auth-token This-is-a-sample-token ^
    --cache-subnet-group sng-test
```

Modification du jeton AUTH sur un cluster existant

Pour faciliter la mise à jour de votre authentification, vous pouvez modifier le AUTH jeton utilisé sur un cluster. Vous pouvez effectuer cette modification si la version du moteur est Valkey 7.2 ou supérieure ou Redis 5.0.6 ou supérieure. ElastiCache le chiffrement en transit doit également être activé.

La modification du jeton d'authentification prend en charge deux stratégies : ROTATE et SET. La stratégie ROTATE ajoute un jeton AUTH supplémentaire au serveur tout en conservant le jeton précédent. La stratégie SET met à jour le serveur pour qu'il ne prenne en charge qu'un seul jeton AUTH. Effectuez ces appels de modification avec le paramètre `--apply-immediately` pour appliquer les modifications immédiatement.

Rotation du jeton AUTH

Pour mettre à jour un serveur Valkey ou Redis OSS avec un nouveau jeton AUTH, appelez l'`ModifyReplicationGroup`API avec le `--auth-token` paramètre comme nouveau AUTH jeton et `--auth-token-update-strategy` avec la valeur ROTATE. Une fois la modification ROTATE terminée, le cluster prendra en charge le jeton AUTH précédent en plus de celui spécifié dans le `auth-token` paramètre. Si aucun jeton AUTH n'a été configuré sur le groupe de réplication avant la rotation du jeton AUTH, le cluster prend en charge le jeton AUTH spécifié dans le `--auth-token` paramètre, en plus de prendre en charge la connexion sans authentification. Voir Définition du jeton AUTH pour mettre à jour le jeton AUTH requis à l'aide de la stratégie de mise à jour SET.

> **Note**
>
> Si vous ne configurez pas le jeton AUTH auparavant, une fois la modification terminée, le cluster ne prendra en charge aucun jeton AUTH en plus de celui spécifié dans le paramètre auth-token.

Si cette modification est effectuée sur un serveur qui prend déjà en charge deux jetons AUTH, le jeton AUTH le plus ancien sera également supprimé lors de cette opération. Cela permet à un serveur de prendre en charge jusqu'à deux jetons AUTH les plus récents à un moment donné.

À ce stade, vous pouvez procéder en mettant à jour le client pour qu'il utilise le dernier jeton AUTH. Une fois les clients mis à jour, vous pouvez utiliser la stratégie SET pour la rotation des jetons AUTH (expliqué dans la section suivante) pour commencer exclusivement à utiliser le nouveau jeton.

L' AWS CLI opération suivante modifie un groupe de réplication pour faire pivoter le AUTH jeton*`This-is-the-rotated-token`*.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-rotated-token \
--auth-token-update-strategy ROTATE \
--apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
```

```
--replication-group-id authtestgroup ^
--auth-token This-is-the-rotated-token ^
--auth-token-update-strategy ROTATE ^
--apply-immediately
```

Définition du jeton AUTH

Pour mettre à jour un serveur Valkey ou Redis OSS afin de prendre en charge un seul AUTH jeton requis, appelez l'opération `ModifyReplicationGroup` API avec le `--auth-token` paramètre ayant la même valeur que le dernier jeton AUTH et le `--auth-token-update-strategy` paramètre avec la valeur. `SET` La stratégie SET ne peut être utilisée qu'avec un cluster doté de 2 jetons AUTH ou d'un jeton AUTH optionnel issu d'une stratégie ROTATE auparavant. Une fois la modification terminée, le serveur ne prend en charge que le jeton AUTH spécifié dans le paramètre auth-token.

L' AWS CLI opération suivante modifie un groupe de réplication pour attribuer au jeton AUTH la valeur. *`This-is-the-set-token`*

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-set-token \
--auth-token-update-strategy SET \
--apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-set-token ^
--auth-token-update-strategy SET ^
--apply-immediately
```

Activation de l'authentification sur un cluster existant

Pour activer l'authentification sur un serveur Valkey ou Redis OSS existant, appelez l'opération `ModifyReplicationGroup` API. Appelez `ModifyReplicationGroup` avec le paramètre `--auth-token` en tant que nouveau jeton et le paramètre `--auth-token-update-strategy` avec la valeur ROTATE.

Une fois la modification ROTATE terminée, le cluster prend en charge le AUTH jeton spécifié dans le `--auth-token` paramètre, en plus de prendre en charge la connexion sans authentification. Une fois que toutes les applications clientes sont mises à jour pour s'authentifier auprès de Valkey ou Redis OSS avec le jeton AUTH, utilisez la stratégie SET pour marquer le jeton AUTH comme requis. L'activation de l'authentification n'est prise en charge que sur les serveurs Valkey et Redis OSS avec le chiffrement en transit (TLS) activé.

Migration de RBAC vers AUTH

Si vous authentifiez des utilisateurs avec Valkey ou Redis OSS Role-Based Access Control (RBAC) comme décrit dansContrôle d'accès basé sur les rôles (RBAC), et que vous souhaitez migrer vers AUTH, suivez les procédures suivantes. Vous pouvez migrer à l'aide de la console ou de la CLI.

Pour migrer de RBAC vers AUTH à l'aide de la console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
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 volet de navigation, choisissez le moteur s'exécutant sur le cluster que vous voulez modifier.

   Une liste de clusters du moteur choisi s'affiche.
4. Dans la liste des clusters, choisissez le nom du cluster que vous souhaitez modifier.
5. Pour Actions, choisissez Modifier.

   La fenêtre Modifier apparaît.
6. Pour le contrôle d'accès, choisissez l'accès utilisateur par défaut Valkey AUTH ou l'accès utilisateur par défaut Redis OSS AUTH.
7. Sous le jeton Valkey AUTH ou le jeton Redis OSS AUTH, définissez un nouveau jeton.
8. Choisissez Prévisualiser les modifications, puis Modifier sur l'écran suivant.

Pour migrer de RBAC vers AUTH à l'aide du AWS CLI

Utilisez l'une des commandes suivantes pour configurer un nouveau AUTH jeton facultatif pour votre groupe de réplication Valkey ou Redis OSS. Notez qu'un jeton d'authentification facultatif autorisera

un accès non authentifié au groupe de réplication jusqu'à ce que le jeton d'authentification soit marqué comme obligatoire, en utilisant la stratégie `SET` de mise à jour décrite à l'étape suivante.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id test \
    --remove-user-groups \
    --auth-token This-is-a-sample-token \
    --auth-token-update-strategy ROTATE \
    --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test ^
    --remove-user-groups ^
    --auth-token This-is-a-sample-token ^
    --auth-token-update-strategy ROTATE ^
    --apply-immediately
```

Après avoir exécuté la commande ci-dessus, vous pouvez mettre à jour vos applications Valkey ou Redis OSS pour vous authentifier auprès du groupe de ElastiCache réplication à l'aide du jeton AUTH optionnel nouvellement configuré. Pour terminer la rotation du jeton d'authentification, utilisez la stratégie de mise à jour décrite `SET` dans la commande suivante ci-dessous. Cela sera marqué sur le jeton AUTH optionnel selon les besoins. Lorsque la mise à jour du jeton d'authentification sera terminée, l'état du groupe de réplication s'affichera comme tel `ACTIVE` et toutes les connexions à ce groupe de réplication nécessiteront une authentification.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
   --replication-group-id test \
   --auth-token This-is-a-sample-token \
   --auth-token-update-strategy SET \
   --apply-immediately
```

Pour Windows :

```
aws elasticache modify-replication-group ^
```

```
    --replication-group-id test ^
    --remove-user-groups ^
    --auth-token This-is-a-sample-token ^
    --auth-token-update-strategy SET ^
    --apply-immediately
```

Pour de plus amples informations, veuillez consulter Authentification avec la commande Valkey et Redis OSS AUTH.

> **Note**
>
> Si vous devez désactiver le contrôle d'accès sur un ElastiCache cluster, consultezthe section called "Désactivation du contrôle d'accès sur un cache ElastiCache Valkey ou Redis OSS".

Désactivation du contrôle d'accès sur un cache ElastiCache Valkey ou Redis OSS

Suivez les instructions ci-dessous pour désactiver le contrôle d'accès sur un cache compatible TLS avec Valkey ou Redis OSS. Votre cache aura l'un des deux types de configuration suivants : accès utilisateur par défaut AUTH ou liste de contrôle d'accès des groupes d'utilisateurs (RBAC). Si votre cache a été créé avec la configuration AUTH, vous devez le remplacer par la configuration RBAC avant de pouvoir désactiver le cache en supprimant les groupes d'utilisateurs. Si votre cache a été créé avec la configuration RBAC, vous pouvez directement le désactiver.

Pour désactiver un cache sans serveur Valkey ou Redis OSS configuré avec RBAC

1. Supprimez les groupes d'utilisateurs pour désactiver le contrôle d'accès.

   ```
   aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
    --remove-user-group
   ```

2. (Facultatif) Vérifiez qu'aucun groupe d'utilisateurs n'est associé au cache sans serveur.

   ```
   aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
   cache>
   {
       "..."
       "UserGroupId": ""
       "..."
   }
   ```

Pour désactiver un cache Valkey ou Redis OSS configuré avec un jeton AUTH

1. Remplacez le jeton AUTH par RBAC et spécifiez le groupe d'utilisateurs à ajouter.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-value>
```

2. Vérifiez que le jeton AUTH a été désactivé et qu'un groupe d'utilisateurs a été ajouté.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-id-value>
{
    "..."
    "AuthTokenEnabled": false,
    "UserGroupIds": [
        "<user-group-value>"
    ]
    "..."
}
```

3. Supprimez les groupes d'utilisateurs pour désactiver le contrôle d'accès.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
    "..."
    "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
    "..."
}
```

4. (Facultatif) Vérifiez qu'aucun groupe d'utilisateurs n'est associé au cluster. Le champ `AuthTokenEnabled` doit également indiquer « false ».

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
```

```
"AuthTokenEnabled": false
```

Pour désactiver un cluster Valkey ou Redis OSS configuré avec RBAC

1. Supprimez les groupes d'utilisateurs pour désactiver le contrôle d'accès.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
value> --user-group-ids-to-remove <user-group-value>
{
    "..."
    "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
    "..."
}
```

2. (Facultatif) Vérifiez qu'aucun groupe d'utilisateurs n'est associé au cluster. Le champ `AuthTokenEnabled` doit également indiquer « false ».

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

Confidentialité du trafic inter-réseau

Amazon ElastiCache utilise les techniques suivantes pour sécuriser les données de votre cache et les protéger contre tout accès non autorisé :

- Amazon VPCs et la ElastiCache sécurité explique le type de groupe de sécurité dont vous avez besoin pour votre installation.
- Identity and Access Management pour Amazon ElastiCache pour attribuer et limiter les actions des utilisateurs, groupes et rôles.

Rubriques

- Amazon VPCs et la ElastiCache sécurité
- ElastiCache Points de terminaison VPC de l'API et de l'interface ()AWS PrivateLink
- Sous-réseaux et groupes de sous-réseaux

Amazon VPCs et la ElastiCache sécurité

Étant donné que la sécurité des données est importante, ElastiCache vous fournit des moyens de contrôler qui a accès à vos données. La manière dont vous contrôlez l'accès à vos données dépend du fait que vous avez lancé ou non vos clusters dans un Amazon Virtual Private Cloud (Amazon VPC) ou Amazon EC2 -Classic.

> **Important**
>
> Nous avons déconseillé l'utilisation d'Amazon EC2 -Classic pour le lancement de clusters. ElastiCache Tous les nœuds de génération actuelle sont lancés dans Amazon Virtual Private Cloud uniquement.

Le service Amazon Amazon Virtual Private Cloud (Amazon VPC) définit un réseau virtuel qui ressemble de près à un centre de données classique. Vous configurez votre Amazon 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. Vous pouvez également ajouter un cluster de cache au réseau virtuel et contrôler l'accès au cluster de cache à l'aide des groupes de sécurité Amazon VPC.

Cette section explique comment configurer manuellement un ElastiCache cluster dans un Amazon VPC. Ces informations sont destinées aux utilisateurs qui souhaitent mieux comprendre comment ElastiCache Amazon VPC fonctionne ensemble.

Rubriques

- Understanding ElastiCache et Amazon VPCs
- Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC
- Création d'un Virtual Private Cloud (VPC)
- Connexion à un cache en cours d'exécution dans un réseau Amazon VPC

Understanding ElastiCache et Amazon VPCs

ElastiCache est entièrement intégré à Amazon Virtual Private Cloud (Amazon VPC). Pour ElastiCache les utilisateurs, cela signifie ce qui suit :

- Si votre AWS compte ne prend en charge que la plate-forme EC2 -VPC, lance ElastiCache toujours votre cluster dans un Amazon VPC.
- Si vous êtes nouveau dans ce AWS domaine, vos clusters seront déployés dans un Amazon VPC. 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 Virtual Private Cloud, vous pouvez créer un réseau virtuel dans le AWS cloud qui ressemble beaucoup à un centre de données traditionnel. Vous pouvez configurer votre Amazon 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.

Les fonctionnalités de base ElastiCache sont les mêmes dans un cloud privé virtuel ; elles ElastiCache gèrent les mises à niveau logicielles, les correctifs, la détection des défaillances et la restauration, que vos clusters soient déployés à l'intérieur ou à l'extérieur d'un Amazon VPC.

ElastiCache les nœuds de cache déployés en dehors d'un Amazon VPC se voient attribuer une adresse IP à laquelle le endpoint/DNS nom est résolu. Cela fournit une connectivité à partir d'instances Amazon Elastic Compute Cloud (Amazon EC2). Lorsque vous lancez un ElastiCache cluster dans un sous-réseau privé Amazon VPC, une adresse IP privée est attribuée à chaque nœud de cache au sein de ce sous-réseau.

Présentation de ElastiCache in a Amazon VPC

Le schéma et le tableau suivants décrivent l'environnement Amazon VPC, ainsi que les ElastiCache clusters et les EC2 instances Amazon lancés dans Amazon VPC.

1	L'Amazon VPC est une partie isolée du AWS cloud à laquelle est attribué son propre bloc d'adresses IP.
2	

	Une passerelle Internet connecte votre Amazon VPC directement à Internet et donne accès à d'autres AWS ressources telles qu'Amazon Simple Storage Service (Amazon S3) qui s'exécutent en dehors de votre Amazon VPC.
3	Un sous-réseau Amazon VPC est un segment de la plage d'adresses IP d'un Amazon VPC dans lequel vous pouvez isoler les AWS ressources en fonction de vos besoins opération nels et de sécurité.
4	Une table de routage dans l'Amazon VPC dirige le trafic réseau entre le sous-réseau et le réseau Internet. L'Amazon VPC a un routeur implicite, qui est symbolisé dans ce diagramme par le cercle avec le R.
5	Un groupe de sécurité Amazon VPC contrôle le trafic entrant et sortant pour vos clusters ElastiCache et instances Amazon. EC2
6	Vous pouvez lancer un ElastiCache cluster dans le sous-réseau. Les nœuds de cache ont des adresses IP privées provenant de la plage d'adresses du sous-réseau.
7	Vous pouvez également lancer des EC2 instances Amazon dans le sous-réseau. Chaque EC2 instance Amazon possède une adresse IP privée issue de la plage d'adresses du sous-réseau. L' EC2 instance Amazon peut se connecter à n'importe quel nœud de cache du même sous-réseau.
8	Pour qu'une EC2 instance Amazon de votre Amazon VPC soit accessible depuis Internet, vous devez lui attribuer une adresse publique statique appelée adresse IP élastique.

Prérequis

Pour créer un ElastiCache cluster au sein d'un Amazon VPC, votre Amazon VPC doit répondre aux exigences suivantes :

- L'Amazon VPC doit autoriser les instances Amazon EC2 non dédiées. Vous ne pouvez pas l'utiliser ElastiCache dans un Amazon VPC configuré pour la location d'instance dédiée.

- Un groupe de sous-réseaux de cache doit être défini pour votre Amazon VPC. ElastiCache utilise ce groupe de sous-réseaux de cache pour sélectionner un sous-réseau et des adresses IP au sein de ce sous-réseau à associer à vos points de terminaison VPC ou à vos nœuds de cache.
- Les blocs CIDR de chaque sous-réseau doivent être suffisamment grands pour fournir des adresses IP de rechange ElastiCache à utiliser lors des activités de maintenance.

Routage et sécurité

Vous pouvez configurer le routage dans votre Amazon VPC pour contrôler le flux du trafic (par exemple, vers la passerelle Internet ou une passerelle réseau privé virtuel). Avec une passerelle Internet, votre Amazon VPC a un accès direct à d'autres AWS ressources qui ne s'exécutent pas dans votre Amazon 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 devez payer des frais de bande passante supplémentaires lorsque vous accédez à AWS des ressources via Internet.

Vous pouvez utiliser les groupes de sécurité Amazon VPC pour sécuriser les ElastiCache clusters et les EC2 instances Amazon de 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 vivement d'utiliser des noms DNS pour vous connecter à vos nœuds de cache, car l'adresse IP sous-jacente peut changer.

Documentation Amazon VPC

Amazon VPC a son propre ensemble de documentation pour décrire comment créer et utiliser votre Amazon VPC. Le tableau suivant indique les liens vers les guides Amazon VPC.

Description	Documentation
Commencer à utiliser Amazon VPC	Mise en route avec Amazon VPC
Comment utiliser Amazon VPC via le AWS Management Console	Amazon VPC User Guide

Description	Documentation
Description complète de toutes les commandes Amazon VPC	Amazon EC2 Command Line Reference (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	Amazon EC2 Command Line Reference (les opérations de l'API Amazon VPC se trouvent dans la référence Amazon EC2)
Informations destinées à l'administrateur réseau qui doit configurer la passerelle à votre extrémité d'une connexion IPsec VPN optionnel le	Qu'est-ce qu' AWS Site-to-Site un VPN ?

Pour plus d'informations sur Amazon Virtual Private Cloud, veuillez consulter Amazon Virtual Private Cloud.

Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC

Amazon ElastiCache prend en charge les scénarios suivants pour accéder à un cache dans un Amazon VPC :

Table des matières

- Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans le même Amazon VPC
- Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans un environnement Amazon différent VPCs
  - Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans un Amazon différent VPCs dans la même région
    - Utilisation de Transit Gateway
  - Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans des régions différentes d'Amazon VPCs
    - Utilisation de VPC en transit
- Accès à un ElastiCache cache depuis une application exécutée dans le centre de données d'un client
  - Accès à un ElastiCache cache depuis une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN
  - Accès à un ElastiCache cache à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans le même Amazon VPC

Le cas d'utilisation le plus courant est celui où une application déployée sur une EC2 instance doit se connecter à un cache dans le même VPC.

Le schéma suivant illustre ce scénario.

Le moyen le plus simple de gérer l'accès entre les EC2 instances et les caches d'un même VPC est de procéder comme suit :

1. Créez un groupe de sécurité VPC pour votre cache. Ce groupe de sécurité peut être utilisé pour restreindre l'accès au cache. 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 cache lorsque vous l'avez créé et d'une adresse IP que vous utiliserez pour accéder au cache.

   Le port par défaut des caches Memcached est `11211`.

   Le port par défaut pour les caches Valkey et Redis OSS est. `6379`
2. Créez un groupe de sécurité VPC pour vos EC2 instances (serveurs Web et d'applications). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l' EC2 instance depuis 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' EC2 instance via le port 22.
3. Créez des règles personnalisées dans le groupe de sécurité de votre cache qui autorisent les connexions depuis le groupe de sécurité que vous avez créé pour vos EC2 instances. N'importe quel membre du groupe de sécurité peut ainsi accéder aux caches.

> **Note**
>
> Si vous envisagez d'utiliser les zones locales, vérifiez que vous les avez activées. Lorsque vous créez un groupe de sous-réseaux dans cette zone locale, votre VPC est étendu à cette zone locale et votre VPC traitera le sous-réseau comme n'importe quel sous-réseau dans

n'importe quelle autre zone de disponibilité. Toutes les passerelles et les tables de routage pertinentes seront automatiquement ajustées.

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 https://console.aws.amazon.com l'adresse /vpc.
2. Dans le panneau de navigation, choisissez Groupes de sécurité.
3. Sélectionnez ou créez un groupe de sécurité que vous utiliserez pour votre cache. 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 cache.

      Le port par défaut des caches Memcached est `11211`.

      Le port par défaut pour les caches et les groupes de réplication Valkey et Redis OSS 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 EC2 instances Amazon.
5. Choisissez Enregistrer lorsque vous avez terminé.

Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans un environnement Amazon différent VPCs

Lorsque votre cache se trouve dans un VPC différent de l' EC2 instance que vous utilisez pour y accéder, il existe plusieurs manières d'accéder au cache. Si le cache et l' EC2 instance se trouvent dans VPCs une région différente, vous pouvez utiliser le peering VPC. Si le cache et l' EC2 instance se trouvent dans des régions différentes, vous pouvez créer une connectivité VPN entre les régions.

Rubriques

- Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans un Amazon différent VPCs dans la même région
- Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans des régions différentes d'Amazon VPCs

Accès à un ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans un Amazon différent VPCs dans la même région

Le schéma suivant illustre l'accès à un cache par une EC2 instance Amazon dans un autre Amazon VPC de la même région à l'aide d'une connexion d'appairage Amazon VPC.

Accès au cache par une EC2 instance Amazon dans un autre Amazon VPC au sein de la même région - VPC Peering Connection

Une connexion d'appairage VPC est une connexion réseau entre deux entités VPCs qui vous permet d'acheminer le trafic entre elles 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 de peering VPC entre votre propre Amazon ou avec un Amazon VPCs 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.

> **Note**
>
> La résolution des noms DNS peut échouer pour peered VPCs, selon les configurations appliquées au ElastiCache VPC. Pour résoudre ce problème, les deux VPCs doivent être activés pour les noms d'hôte DNS et la résolution DNS. Pour plus d'informations, consultez Activation de la résolution DNS pour une connexion d'appairage de VPC.

Pour accéder à un cache dans un réseau Amazon VPC différent via l'appairage

1. Assurez-vous que les deux adresses IP VPCs ne se chevauchent pas, sinon vous ne pourrez pas les comparer.
2. Regardez les deux VPCs. 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.

   Voici à quoi ressemble les tables de routage pour l'exemple du schéma précédent. Notez que pcx-a894f1c1 correspond à la connexion d'appairage.

Destination	Target
172.16.0.0/16	local
10.10.0.0/16	pcx-a894f1c1

Destination	Target
10.10.0.0/16	local
0.0.0.0/0	igw-bfdcccd8
172.16.0.0/16	pcx-a894f1c1

   Table de routage VPC

4. Modifiez le groupe de sécurité de votre ElastiCache cache 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 cache via une connexion d'appairage entraînera des frais de transfert de données supplémentaires.

Utilisation de Transit Gateway

Une passerelle de transit vous permet de connecter VPCs des connexions VPN dans la même AWS région et d'acheminer le trafic entre elles. 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 titulaire du compte peut l'associer VPCs 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 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 ElastiCache cache lorsque celui-ci et l' EC2 instance Amazon se trouvent dans des régions différentes d'Amazon VPCs

Utilisation de VPC en transit

Une alternative à l'utilisation du peering VPC, une autre stratégie courante pour connecter plusieurs réseaux distants VPCs et géographiquement dispersés consiste à créer un VPC de transit faisant office de centre de transit du réseau mondial. Un VPC de transit simplifie la gestion du réseau et minimise le nombre de connexions nécessaires pour connecter plusieurs réseaux VPCs 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 entre VPCs différentes régions

Une fois le VPC Transit Amazon établi, une application déployée dans un VPC « relais » d'une région peut se connecter à un ElastiCache cache d'un VPC « relais » d'une autre région.

Pour accéder à un cache 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 le cache VPCs 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 ElastiCache cache 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 cache d'une région à une autre entraînera des latences réseau et des frais supplémentaires de transfert de données entre régions.

Accès à un ElastiCache cache depuis 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 ElastiCache cache 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 ElastiCache cache depuis une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN
- Accès à un ElastiCache cache à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Accès à un ElastiCache cache depuis une application exécutée dans le centre de données d'un client à l'aide de la connectivité VPN

Le schéma suivant illustre l'accès à un ElastiCache cache à partir d'une application exécutée sur le réseau de votre entreprise à l'aide de connexions VPN.

Connexion ElastiCache depuis votre centre de données via un VPN

Pour accéder à un cache 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 ElastiCache cache 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 ElastiCache cache pour autoriser les connexions entrantes depuis les serveurs d'applications locaux.

Le fait d'accéder à un cache via une connexion VPN entraînera des latences réseau et des frais de transfert de données supplémentaires.

Accès à un ElastiCache cache à partir d'une application exécutée dans le centre de données d'un client à l'aide de Direct Connect

Le schéma suivant illustre l'accès à un ElastiCache cache à partir d'une application exécutée sur le réseau de votre entreprise à l'aide de Direct Connect.

Connexion ElastiCache depuis votre centre de données via Direct Connect

Pour accéder à un ElastiCache cache depuis 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 ElastiCache cache pour autoriser les connexions entrantes depuis les serveurs d'applications locaux.

Le fait d'accéder à un cache 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 Amazon VPC avec un sous-réseau privé pour chaque zone de disponibilité.

Création d'un Amazon VPC (console)

1. Connectez-vous à la console AWS de gestion et ouvrez la console Amazon VPC à l'adresse. https://console.aws.amazon.com/vpc/
2. Dans le tableau de bord VPC, choisissez Create VPC (Créer un VPC).
3. Sous Resources to create (Ressources à créer), choisissez VPC and more (VPC et autres).
4. Sous Nombre de zones de disponibilité (AZs), choisissez le nombre de zones de disponibilité dans lesquelles vous souhaitez lancer vos sous-réseaux.
5. Sous Number of public subnets (Nombre de sous-réseaux publics), choisissez le nombre de sous-réseaux publics que vous voulez ajouter à votre VPC.
6. Sous Number of private subnets (Nombre de sous-réseaux privés), choisissez le nombre de sous-réseaux privés que vous voulez ajouter à votre VPC.

   > **Tip**
   >
   > Notez les identifiants de vos sous-réseaux, et notez lesquels sont publics et privés. Vous aurez besoin de ces informations ultérieurement lorsque vous lancerez vos clusters et ajouterez une EC2 instance Amazon à votre Amazon VPC.

7. Créez un groupe de sécurité Amazon VPC. Vous utiliserez ce groupe pour votre cluster de cache et votre EC2 instance Amazon.

   a. Dans le panneau de navigation de la console de gestion Amazon VPC, choisissez Security Groups (Groupes de sécurité).

   b. Sélectionnez Create Security Group (Créer un groupe de sécurité).

   c. Tapez un nom et une description pour votre groupe de sécurité dans les zones correspondantes. Dans la zone VPC, choisissez l'identifiant de votre Amazon 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 vous permettra de vous connecter à votre EC2 instance Amazon à l'aide de Secure Shell (SSH).

   a. Dans la liste de navigation, choisissez 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.
   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.

Vous êtes maintenant prêt à créer un groupe de sous-réseaux de cache et à lancer un cluster de cache dans votre Amazon VPC.

- Création d'un groupe de sous-réseaux
- Création d'un cluster Memcached (console).
- Création d'un cluster Valkey (mode cluster désactivé) (console).

Connexion à un cache en cours d'exécution dans un réseau Amazon VPC

Cet exemple montre comment lancer une EC2 instance Amazon dans votre Amazon VPC. Vous pouvez ensuite vous connecter à cette instance et accéder au ElastiCache cache qui s'exécute dans Amazon VPC.

Connexion à un cache en cours d'exécution dans un réseau Amazon VPC (console)

Dans cet exemple, vous créez une EC2 instance Amazon dans votre Amazon VPC. Vous pouvez utiliser cette EC2 instance Amazon pour vous connecter aux nœuds de cache exécutés dans Amazon VPC.

> **Note**
>
> Pour plus d'informations sur l'utilisation d'Amazon EC2, consultez le guide de EC2 démarrage Amazon dans la EC2 documentation Amazon.

Pour créer une EC2 instance Amazon dans votre Amazon VPC à l'aide de la console Amazon EC2

1. Connectez-vous à la EC2 console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans la console, sélectionnez Lancer l'instance et suivez les étapes ci-dessous :
3. Sur la page Choisissez une Amazon Machine Image (AMI), choisissez une AMI d'Amazon Linux 64 bits, puis Sélectionner.
4. Sur la page Choisir un type d'instance, choisissez 3. Configurer l'Instance.
5. Dans la page Configurer les détails de l'instance , effectuez les sélections suivantes :
   a. Dans la liste Network (Réseau), sélectionnez votre Amazon VPC.
   b. Dans la liste Sous-réseau, sélectionnez votre sous-réseau public.

Lorsque les paramètres vous conviennent, choisissez 4. Ajouter le stockage.

6. Sur la page Ajouter du stockage, choisissez 5. Etiqueter l'Instance.
7. Sur la page Tag Instance, saisissez le nom de votre EC2 instance Amazon, puis choisissez 6. Configurer le groupe de sécurité.
8. Sur la page Configurez le groupe de sécurité, choisissez Sélectionnez un groupe de sécurité existant. Pour plus d'informations sur les groupes de sécurité, consultez la section Groupes EC2 de sécurité Amazon pour les instances Linux.

Choisissez le nom du groupe de sécurité de votre Amazon VPC, puis choisissez Review and Launch (Revoir et lancer).

9. Sur la page Vérifier l'instance et lancer, choisissez Lancer.
10. Dans la fenêtre Select an existing key pair or create a new key pair (Sélectionner une paire de clés existante ou créez une paire de clés), spécifiez une paire de clés que vous voulez associer à cette instance.

> **Note**
>
> Pour plus d'informations sur la gestion des paires de clés, consultez le guide de EC2 démarrage Amazon.

11. Lorsque vous êtes prêt à lancer votre EC2 instance Amazon, choisissez Launch.

Vous pouvez désormais attribuer une adresse IP élastique à l' EC2 instance Amazon que vous venez de créer. Vous devez utiliser cette adresse IP pour vous connecter à l' EC2 instance Amazon.

Pour affecter une adresse IP élastique (Console)

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans la liste de navigation, sélectionnez Elastic IPs.
3. Choisissez Allocate Elastic IP address (Allouer l'adresse IP Elastic).
4. Dans la boîte de dialogue Allouer l'adresse IP Elastic, acceptez le Groupe de bordure réseau par défaut, puis Allouer .
5. Choisissez l'adresse IP Elastic que vous venez d'allouer dans la liste, puis choisissez Associer l'adresse.
6. Dans la boîte de dialogue Adresse associée, dans la zone Instance, choisissez l'ID de l' EC2 instance Amazon que vous avez lancée.

   Dans la zone Adresse IP privée cochez la case pour obtenir l'adresse IP privée, puis choisissez Associer.

   Vous pouvez désormais utiliser SSH pour vous connecter à l' EC2 instance Amazon à l'aide de l'adresse IP élastique que vous avez créée.

Pour vous connecter à votre EC2 instance Amazon

- Ouvrez une fenêtre de commande. A l'invite de commande, exécutez la commande suivante, en remplaçant mykeypair.pem par le nom de votre fichier de paire de clés et 54.207.55.251 par votre adresse IP Elastic.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

> **Important**
>
> Ne vous déconnectez pas encore de votre EC2 instance Amazon.

Vous êtes maintenant prêt à interagir avec votre ElastiCache cluster. Avant de pouvoir effectuer cela, si vous n'avez pas déjà fait, vous devez installer l'utilitaire telnet.

Pour installer telnet et interagir avec votre cluster de cache (AWS CLI)

- Ouvrez une fenêtre de commande. À l'invite de commande, émettez la commande suivante. A l'invite de confirmation, tapez y.

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                        |  63 kB     00:00

...(output omitted)...

Complete!
```

Vous pouvez désormais vous connecter à un VPC avec Memcached ou Redis.

Connexion à un VPC avec Memcached

1. Accédez à la ElastiCache console https://console.aws.amazon.com/elasticache/et obtenez le point de terminaison de l'un des nœuds de votre cluster de cache. Pour plus d'informations, veuillez consulter Recherche de points de terminaison de connexion.

2. Utilisez telnet pour vous connecter à votre point de terminaison du nœud de cache via le port 11211. Remplacez le nom d'hôte indiqué ci-dessous par le nom d'hôte de votre nœud de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Vous êtes actuellement connecté au moteur de cache et vous pouvez envoyer des commandes. Dans cet exemple, vous ajoutez un élément de données au cache, puis vous l'obtenez immédiatement après. Enfin, vous allez vous déconnecter du nœud de cache.

Pour stocker une clé et une valeur, saisissez les deux lignes suivantes :

```
add mykey 0 3600 28
This is the value for mykey
```

Le moteur de cache répond avec ce qui suit :

```
OK
```

Pour récupérer la valeur de `mykey`, saisissez les données ci-dessous :

```
get mykey
```

Le moteur de cache répond avec ce qui suit :

```
VALUE mykey 0 28
This is the value for my key
END
```

Pour vous déconnecter du moteur de cache, saisissez ce qui suit :

```
quit
```

Connexion à un VPC avec Redis

1. Accédez à la ElastiCache console https://console.aws.amazon.com/elasticache/et obtenez le point de terminaison de l'un des nœuds de votre cluster de cache. Pour plus d'informations, consultez la section Recherche de points de terminaison de connexion pour Redis.

2. Utilisez telnet pour vous connecter à votre point de terminaison du nœud de cache via le port 6379. Remplacez le nom d'hôte indiqué ci-dessous par le nom d'hôte de votre nœud de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Vous êtes actuellement connecté au moteur de cache et vous pouvez envoyer des commandes. Dans cet exemple, vous ajoutez un élément de données au cache, puis vous l'obtenez immédiatement après. Enfin, vous allez vous déconnecter du nœud de cache.

Pour stocker une clé et une valeur, saisissez les deux lignes suivantes :

```
set mykey myvalue
```

Le moteur de cache répond avec ce qui suit :

```
OK
```

Pour récupérer la valeur de `mykey`, saisissez les données ci-dessous :

```
get mykey
```

Pour vous déconnecter du moteur de cache, saisissez ce qui suit :

```
quit
```

3. Accédez à la ElastiCache console https://console.aws.amazon.com/elasticache/et obtenez le point de terminaison de l'un des nœuds de votre cluster de cache. Pour plus d'informations, voir Recherche de points de terminaison de connexion pour Redis OSS.
4. Utilisez telnet pour vous connecter à votre point de terminaison du nœud de cache via le port 6379. Remplacez le nom d'hôte indiqué ci-dessous par le nom d'hôte de votre nœud de cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Vous êtes actuellement connecté au moteur de cache et vous pouvez envoyer des commandes. Dans cet exemple, vous ajoutez un élément de données au cache, puis vous l'obtenez immédiatement après. Enfin, vous allez vous déconnecter du nœud de cache.

Pour stocker une clé et une valeur, saisissez ce qui suit :

```
set mykey myvalue
```

Le moteur de cache répond avec ce qui suit :

```
OK
```

Pour récupérer la valeur de `mykey`, saisissez les données ci-dessous :

```
get mykey
```

Le moteur de cache répond avec ce qui suit :

```
get mykey
myvalue
```

Pour vous déconnecter du moteur de cache, saisissez ce qui suit :

```
quit
```

> **Important**
>
> Pour éviter d'encourir des frais supplémentaires sur votre AWS compte, veillez à supprimer les AWS ressources dont vous n'avez plus besoin après avoir essayé ces exemples.

ElastiCache Points de terminaison VPC de l'API et de l'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et les points de terminaison d' ElastiCache API Amazon 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 ElastiCache l'API Amazon sans passerelle Internet, appareil NAT, connexion VPN ou connexion AWS Direct Connect.

Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les points de terminaison des ElastiCache API Amazon. Vos instances n'ont pas non plus besoin d'adresses IP publiques pour utiliser les opérations d' ElastiCache API disponibles. Le trafic entre votre VPC et Amazon ElastiCache 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 élastiques, consultez la section relative aux interfaces réseau élastiques dans le guide de EC2 l'utilisateur Amazon.

- 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 ElastiCache d'API, consultez la section Opérations d'ElastiCache API.

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 ElastiCache terminaison par défaut (https://elasticache). *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.elasticache.Region.vpce.amazonaws.com
```

Pour plus d'informations, consultez Interface VPC Endpoints (AWS PrivateLink) dans le guide de l'utilisateur Amazon VPC. ElastiCache 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 souhaitez créer un point de terminaison VPC supplémentaire, le nom d'hôte DNS privé doit être désactivé pour ce point de terminaison.

Régions Privatelink disponibles

Code	Emplacement	Région
CPT	Afrique (Le Cap)	AF-SOUTH-1

Code	Emplacement	Région
HKG	Asie-Pacifique (Hong Kong)	AP-EST-1
TPE	Asie-Pacifique (Taipei)	AP-EST-2
NRT	Asie-Pacifique (Tokyo)	AP-NORTHEAST-1
ICN	Asie-Pacifique (Séoul)	AP-NORTHEAST-2
KIX	Asie-Pacifique (Osaka)	AP-NORTHEAST-3
Nomenclature	Asie-Pacifique (Mumbai)	AP-SOUTH 1
HYD	Asie-Pacifique (Hyderabad)	AP-SOUTH 2
SIN	Asie-Pacifique (Singapour)	AP-SOUTHEAST-1
SYD	Asie-Pacifique (Sydney)	AP-SOUTHEAST-2
CGK	Asie-Pacifique (Jakarta)	AP-SOUTHEAST-3
MEL	Asie-Pacifique (Melbourne)	AP-SOUTHEAST-4
KUL	Asie-Pacifique (Malaisie)	AP-SOUTHEAST-5
BKK	Asie-Pacifique (Thaïlande)	AP-SOUTHEAST-7
YUL	Canada (Centre)	CA-CENTRAL-1
YYC	Canada Ouest (Calgary)	CA-OUEST-1
BJS	Chine (Beijing)	CN-NORTH-1
ZHY	Chine (Ningxia)	CN-NORTHWEST-1
FRA	Europe (Francfort)	EU-CENTRAL-1
ZRH	Europe (Zurich)	EU-CENTRAL-2
ARN	Europe (Stockholm)	EU-NORD-1

Code	Emplacement	Région
MXP	Europe (Milan)	EU-SUD-1
ZAZ	Europe (Espagne)	EU-SUD-2
DUB	Europe (Irlande)	EU-OUEST-1
LHR	Europe (Londres)	EU-OUEST-2
CDG	Europe (Paris)	EU-OUEST-3
TLV	Tel Aviv (Israël)	IL-CENTRAL-1
DXB	Moyen-Orient (EAU)	ME-CENTRAL-1
BAH	Moyen-Orient (Bahreïn)	MOI-SUD-1
QRO	Mexique (centre)	MX-CENTRAL-1
GRU	Amérique du Sud (Sao Paulo)	SA-EAST-1
IAD	USA Est (Virginie du Nord)	US-EAST-1
CMH	USA Est (Ohio)	US-EAST-2
OSU	AWS GovCloud (USA Est)	US-GOV-EAST-1
SFO	USA Ouest (Californie du Nord)	US-OUEST-1
PDX	USA Ouest (Oregon)	US-OUEST-2
PDT	AWS GovCloud (US-Ouest)	US-OUEST-1

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' ElastiCache API Amazon, 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 ElastiCache

d'API relatives à la gestion des ElastiCache ressources Amazon 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 d' ElastiCache API. Par défaut, l'accès complet aux opérations de ElastiCache l'API 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 ElastiCache

Vous pouvez créer un point de terminaison VPC pour l' ElastiCache API Amazon 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 d'un VPC d'interface, vous pouvez activer les noms d'hôte DNS privés pour le point de terminaison. Lorsque vous le faites, le point de ElastiCache terminaison Amazon par défaut (https://elasticache.*Region*.amazonaws.com) correspond à votre point de terminaison VPC. Pour les AWS régions Chine (Pékin) et Chine (Ningxia), vous pouvez effectuer des demandes d'API avec le point de terminaison VPC `elasticache.cn-north-1.amazonaws.com.cn` en utilisant pour Pékin `elasticache.cn-northwest-1.amazonaws.com.cn` et pour Ningxia. 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 ElastiCache

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à l' ElastiCache API. 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 d' ElastiCache API avec Valkey ou Redis OSS

Voici un exemple de politique de point de terminaison pour l' ElastiCache API. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux actions d' ElastiCache API répertoriées à tous les principaux sur toutes les ressources.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "elasticache:CreateCacheCluster",
   "elasticache:ModifyCacheCluster",
   "elasticache:CreateSnapshot"
  ],
  "Resource": "*"
 }]
}
```

Example Politique de point de terminaison VPC ElastiCache pour les actions d'API Memcached

Voici un exemple de politique de point de terminaison pour l' ElastiCache API. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux actions d' ElastiCache API répertoriées à tous les principaux sur toutes les ressources.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "elasticache:CreateCacheCluster",
   "elasticache:ModifyCacheCluster"
  ],
  "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"
    ]
   }
  }
 ]
}
```

Sous-réseaux et groupes de sous-réseaux

Un groupe de sous-réseaux est un ensemble de sous-réseaux (généralement privés) que vous pouvez indiquer pour vos clusters auto-conçus s'exécutant dans un environnement Amazon Virtual Private Cloud (VPC).

Si vous créez un cluster conçu par vos soins dans un Amazon VPC, vous devez utiliser un groupe de sous-réseaux. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds.

ElastiCache fournit un groupe de IPv4 sous-réseaux par défaut ou vous pouvez choisir d'en créer un nouveau. En IPv6 effet, vous devez créer un groupe de sous-réseaux avec un bloc IPv6 CIDR. Si vous choisissez la double pile, vous devez sélectionner un type d'IP de découverte, IPv6 soit IPv4.

ElastiCache Serverless n'utilise pas de ressource de groupe de sous-réseaux, mais utilise une liste de sous-réseaux directement lors de la création.

Cette section explique comment créer et exploiter des sous-réseaux et des groupes de sous-réseaux pour gérer l'accès à vos ElastiCache ressources.

Pour plus d'informations sur l'utilisation du groupe de sous-réseaux dans un environnement Amazon VPC, veuillez consulter Accès à votre ElastiCache cluster ou à votre groupe de réplication.

Rubriques

- Création d'un groupe de sous-réseaux
- Attribution d’un groupe de sous-réseaux à un cache
- Modification d'un groupe de sous-réseaux
- Suppression d'un groupe de sous-réseaux

Création d'un groupe de sous-réseaux

Un groupe de sous-réseaux de cache est un ensemble de sous-réseaux que vous souhaitez indiquer dans vos caches dans un VPC. Lors du lancement d'un cache dans un VPC, vous devez sélectionner un groupe de sous-réseaux de cache. ElastiCache Utilise ensuite ce groupe de sous-réseaux de cache pour attribuer des adresses IP au sein de ce sous-réseau à chaque nœud de cache du cache.

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 à un 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.

Si vous choisissez IPV4 le type de réseau, un groupe de sous-réseaux par défaut sera disponible ou vous pouvez choisir d'en créer un nouveau. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et les adresses IP de ce sous-réseau à associer à vos nœuds. Si vous choisissez la double pile ou IPV6, vous serez invité à créer une double pile ou IPV6 des sous-réseaux. Pour plus d'informations sur les types de réseau, consultez Choix d'un type de réseau dans ElastiCache. Pour plus d'informations, consultez Création d'un sous-réseau dans votre VPC.

Les procédures suivantes vous montrent comment créer un groupe de sous-réseaux appelé `mysubnetgroup` (console), le AWS CLI, et l'ElastiCache API.

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 AWS de gestion et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.
2. Dans la liste de navigation, choisissez Groupes de sous-réseaux.
3. Choisissez Créer groupe de sous-réseaux.
4. Dans l'assistant Créer un groupe de sous-réseaux, procédez comme suit. Lorsque tous les paramètres vous conviennent, choisissez Créer.

    a. Dans le champ Name, saisissez le nom de votre groupe de sous-réseaux de
    b. Dans la zone Description, saisissez une description de votre groupe de sous-réseaux de

c. Dans la zone ID de VPC, choisissez votre réseau Amazon VPC.

d. Tous les sous-réseaux sont choisis par défaut. Dans le panneau Sous-réseaux sélectionnés, cliquez sur Gérer et sélectionnez les zones de disponibilité ou les zones locales ainsi IDs que vos sous-réseaux privés, puis choisissez Choisir.

5. 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 ElastiCache console. 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-cache-subnet-group` pour créer un groupe de sous-réseaux de

Pour Linux, macOS ou Unix :

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "Testing" \
    --subnet-ids subnet-53df9c3a
```

Pour Windows :

```
aws elasticache create-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-37c3cd17",
        "CacheSubnetGroupDescription": "Testing",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-53df9c3a",
                "SubnetAvailabilityZone": {
```

```
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubriquecreate-cache-subnet-group.

Attribution d'un groupe de sous-réseaux à un cache

Une fois que vous avez créé un groupe de sous-réseaux, vous pouvez lancer un cache dans un réseau Amazon VPC. Pour plus d'informations, consultez les rubriques suivantes.

- Cluster Memcached : pour lancer un cluster Memcached, veuillez consulter Création d'un cluster Memcached (console). A l'étape 7.a (Paramètres memcached avancés), choisissez un groupe de sous-réseaux VPC.
- Cluster autonome Valkey ou Redis OSS — Pour lancer un cluster Valkey ou Redis OSS à nœud unique, voir. Création d'un cluster Valkey (mode cluster désactivé) (console) À l'étape 7.a (Paramètres avancés de Redis OSS), choisissez un groupe de sous-réseaux VPC.
- Groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) — Pour lancer un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) dans un VPC, consultez. Création d'un groupe de réplication Valkey ou Redis OSS (mode cluster désactivé) à partir de zéro À l'étape 7.b (Paramètres avancés de Redis OSS), choisissez un groupe de sous-réseaux VPC.
- Groupe de réplication Valkey ou Redis OSS (mode cluster activé) —. Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console) À l'étape 6.i (Paramètres avancés de Redis OSS), choisissez un groupe de sous-réseaux VPC.

Modification d'un groupe de sous-réseaux

Vous pouvez modifier la description d'un groupe de sous-réseaux ou modifier la liste des sous-réseaux IDs associés au groupe de sous-réseaux. Vous ne pouvez pas supprimer un ID de sous-réseau d'un groupe de sous-réseaux si un cache utilise actuellement ce sous-réseau.

Les procédures suivantes vous montrent comment modifier un groupe de sous-réseaux.

Modification de groupes de sous-réseaux (Console)

Pour modifier un groupe de sous-réseaux

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux).
3. Dans la liste des groupes de sous-réseaux, sélectionnez le bouton d'option de celui que vous voulez modifier, puis choisissez Modifier.
4. Dans le panneau Sous-réseaux sélectionnés, choisissez Gérer.
5. Apportez des modifications aux sous-réseaux sélectionnés, puis cliquez sur Choisir.
6. Cliquez sur Enregistrer les modifications pour enregistrer vos modifications.

Modification de groupes de sous-réseaux (AWS CLI)

A l'invite de commande, utilisez la commande `modify-cache-subnet-group` pour modifier un groupe de sous-réseaux de

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup \
    --cache-subnet-group-description "New description" \
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Pour Windows :

```
aws elasticache modify-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup ^
    --cache-subnet-group-description "New description" ^
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Cette commande doit produire une sortie similaire à ce qui suit :

```
{
    "CacheSubnetGroup": {
        "VpcId": "vpc-73cd3c17",
        "CacheSubnetGroupDescription": "New description",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-42dcf93a",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            },
            {
                "SubnetIdentifier": "subnet-48fc12a9",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ],
        "CacheSubnetGroupName": "mysubnetgroup"
    }
}
```

Pour plus d'informations, consultez la AWS CLI rubriquemodify-cache-subnet-group.

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 s'il est actuellement utilisé par un cache.

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 ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux).
3. Dans la liste des groupes de sous-réseaux, choisissez celui que vous souhaitez supprimer, puis choisissez Supprimer.
4. Lorsque vous êtes invité à confirmer cette opération, entrez le nom du groupe de sous-réseaux dans la zone de saisie de texte, puis choisissez Supprimer.

Suppression d'un groupe de sous-réseaux (AWS CLI)

À l'aide de AWS CLI, appelez la commande delete-cache-subnet-group avec le paramètre suivant :

- `--cache-subnet-group-name` *`mysubnetgroup`*

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name mysubnetgroup
```

Pour Windows :

```
aws elasticache delete-cache-subnet-group ^
    --cache-subnet-group-name mysubnetgroup
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la AWS CLI rubriquedelete-cache-subnet-group.

Identity and Access Management pour Amazon ElastiCache

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser ElastiCache les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l'aide de politiques
- Comment Amazon ElastiCache travaille avec IAM
- Exemples de politiques basées sur l'identité pour Amazon ElastiCache
- Résolution des problèmes d' ElastiCache identité et d'accès à Amazon
- Contrôle d'accès
- Vue d'ensemble de la gestion des autorisations d'accès à vos ElastiCache ressources

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. ElastiCache

Utilisateur du service : si vous utilisez le ElastiCache service 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 nouvelles ElastiCache fonctionnalités 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 ElastiCache, consultez Résolution des problèmes d' ElastiCache identité et d'accès à Amazon.

Administrateur du service — Si vous êtes responsable des ElastiCache ressources de votre entreprise, vous avez probablement un accès complet à ElastiCache. C'est à vous de déterminer les ElastiCache fonctionnalités et les ressources 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 manière dont votre entreprise peut utiliser IAM avec ElastiCache, voirComment Amazon ElastiCache travaille avec IAM.

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaiterez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à ElastiCache. Pour consulter des exemples de politiques ElastiCache basées sur l'identité que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour Amazon ElastiCache

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 des demandes vous-même, consultez AWS Signature Version 4 pour les demandes 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 plus

d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant des informations d'identification d'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des

mots de passe et des 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 d'effectuer une rotation des 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 nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM 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. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). 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 Méthodes pour endosser un rôle 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 Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions 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 un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des

clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour 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 exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon 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 Définition d'autorisations IAM personnalisées avec des politiques gérées par le client 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. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste 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 annule 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 (SCPs) : SCPs 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 des politiques de contrôle des services (SCPs) 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 Organizations et consultez SCPs les politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.
- Politiques de contrôle des ressources (RCPs) : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- 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 session 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 Amazon ElastiCache travaille avec IAM

Avant d'utiliser IAM pour gérer l'accès à ElastiCache, découvrez les fonctionnalités IAM disponibles. ElastiCache

Fonctionnalités IAM que vous pouvez utiliser avec Amazon ElastiCache

Fonctionnalité IAM	ElastiCache soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
ACLs	Oui
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles de service	Oui

Fonctionnalité IAM	ElastiCache soutien
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont ElastiCache les autres AWS services fonctionnent 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 ElastiCache

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 Définition d'autorisations IAM personnalisées avec des politiques gérées par le client 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 ElastiCache

Pour consulter des exemples de politiques ElastiCache basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon ElastiCache

Politiques basées sur les ressources dans ElastiCache

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. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les

utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal intercompte 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 Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Actions politiques pour ElastiCache

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des ElastiCache actions, consultez la section Actions définies par Amazon ElastiCache dans le Service Authorization Reference.

Les actions de politique en ElastiCache cours utilisent le préfixe suivant avant l'action :

```
elasticache
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "elasticache:action1",
      "elasticache: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": "elasticache:Describe*"
```

Pour consulter des exemples de politiques ElastiCache basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon ElastiCache

Ressources politiques pour ElastiCache

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de ElastiCache ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par Amazon ElastiCache dans le Service Authorization Reference.

Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon ElastiCache.

Pour consulter des exemples de politiques ElastiCache basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon ElastiCache

Clés de condition de politique pour ElastiCache

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de ElastiCache condition, consultez la section Clés de condition pour Amazon ElastiCache dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon ElastiCache.

Pour consulter des exemples de politiques ElastiCache basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon ElastiCache

Listes de contrôle d'accès (ACLs) dans ElastiCache

Supports ACLs : Oui

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs 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 ElastiCache

Prise en charge d'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 ABAC, consultez Définition d'autorisations avec l'autorisation 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 ElastiCache

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Passage d'un rôle utilisateur à un rôle IAM (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations de principal interservices pour ElastiCache

Prend en charge les sessions d'accès direct (FAS) : oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour ElastiCache

Prend en charge les rôles de service : oui

Un rôle 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 un rôle de service à partir d'IAM. Pour plus

d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber ElastiCache les fonctionnalités. Ne modifiez des rôles de service que quand ElastiCache vous le conseille.

Rôles liés à un service pour ElastiCache

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. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

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 Amazon ElastiCache

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources ElastiCache. 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 IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par ElastiCache, y compris le format de ARNs pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon ElastiCache dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la ElastiCache console
- 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 ElastiCache des ressources 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.
- Accordez 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 d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez 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 Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM 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 politiques avec 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 la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec 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 ElastiCache console

Pour accéder à la ElastiCache console Amazon, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails ElastiCache des ressources de votre Compte AWS. Si vous créez une politique 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 politique.

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 ElastiCache console, associez également la politique ElastiCache `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' ElastiCache identité et d'accès à Amazon

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec ElastiCache IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans ElastiCache
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ElastiCache ressources

Je ne suis pas autorisé à effectuer une action dans ElastiCache

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 `elasticache:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 elasticache: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 `elasticache:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à ElastiCache.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans ElastiCache. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction 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 ElastiCache ressources

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 (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises ElastiCache en charge, consultezComment Amazon ElastiCache travaille 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 en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Contrôle d'accès

Vous pouvez disposer d'informations d'identification valides pour authentifier vos demandes, mais vous ne pouvez pas créer de ElastiCache ressources ou y accéder sans autorisation. Par exemple, vous devez disposer des autorisations nécessaires pour créer un ElastiCache cluster.

Les sections suivantes décrivent comment gérer les autorisations pour ElastiCache. Nous vous recommandons de lire d'abord la présentation.

- Vue d'ensemble de la gestion des autorisations d'accès à vos ElastiCache ressources
- Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon ElastiCache

Vue d'ensemble de la gestion des autorisations d'accès à vos ElastiCache ressources

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, Amazon prend ElastiCache également en charge l'ajout 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. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le 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 dans le 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

- ElastiCache Ressources et opérations d'Amazon
- Présentation de la propriété des ressources

- Gestion de l'accès aux ressources
- AWS politiques gérées pour Amazon ElastiCache
- Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon ElastiCache
- Autorisations de niveau ressource
- Utilisation de clés de condition
- Utilisation de rôles liés à un service pour Amazon ElastiCache
- ElastiCache Autorisations d'API : référence aux actions, aux ressources et aux conditions

ElastiCache Ressources et opérations d'Amazon

Pour consulter la liste des types de ElastiCache ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par Amazon ElastiCache dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon ElastiCache.

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 racine, 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 de cache. Dans ce cas, votre AWS compte est le propriétaire de la ressource. Dans ElastiCache, la ressource est le cluster de cache.
- Supposons que vous créiez un utilisateur IAM dans votre AWS compte et que vous accordiez des autorisations pour créer un cluster de cache à cet utilisateur. Dans ce cas, l'utilisateur peut créer un cluster de cache. Toutefois, votre AWS compte, auquel appartient l'utilisateur, est propriétaire de la ressource du cluster de cache.
- Supposons que vous créiez un rôle IAM dans votre AWS compte avec les autorisations nécessaires pour créer un cluster de cache. Dans ce cas, toute personne pouvant endosser le rôle peut créer un cluster de cache. Votre AWS compte, auquel appartient le rôle, possède la ressource du cluster de cache.

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 d'Amazon ElastiCache. 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 ElastiCache ressource, telle qu'un cluster de cache, 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'`DescribeCacheClusters`action pour votre AWS compte. ElastiCache permet également d'identifier des ressources spécifiques à l'aide de la ressource ARNs pour les actions d'API. (Cette approche est également appelée autorisations au niveau des ressources.)

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DescribeCacheClusters",
            "Effect": "Allow",
            "Action": [
                "elasticache:DescribeCacheClusters"
            ],
            "Resource": "resource-arn"
        }
    ]
}
```

Pour plus d'informations sur l'utilisation de politiques basées sur l'identité avec ElastiCache, consultez. Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon ElastiCache 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 ElastiCache ressource Amazon (voir ElastiCache Ressources et opérations d'Amazon), le service définit un ensemble d'opérations d'API (voir Actions). Pour accorder des autorisations pour ces opérations d'API ElastiCache , définissez un ensemble d'actions que vous pouvez spécifier dans une politique. Par exemple, pour la ressource de ElastiCache cluster, les actions suivantes sont définies : `CreateCacheClusterDeleteCacheCluster`, et`DescribeCacheCluster`. 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 de plus amples informations, veuillez consulter ElastiCache Ressources et opérations d'Amazon.
- 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'`elasticache:CreateCacheCluster`autorisation autorise ou refuse à l'utilisateur l'autorisation d'effectuer l' ElastiCache`CreateCacheCluster`opération Amazon.
- 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 consulter un tableau présentant toutes les actions de ElastiCache l'API Amazon, consultezElastiCache Autorisations d'API : 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.

Pour exprimer des conditions, vous utilisez des clés de condition prédéfinies. Pour utiliser des clés de condition ElastiCache spécifiques, consultezUtilisation de clés de condition. Il existe des AWS clés de condition larges que vous pouvez utiliser le cas échéant. Pour obtenir la liste complète des clés AWS-wide, consultez la section Clés disponibles pour les conditions dans le guide de l'utilisateur IAM.

AWS politiques gérées pour Amazon ElastiCache

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS politique gérée : ElastiCacheServiceRolePolicy

Vous ne pouvez pas vous associer ElastiCacheServiceRolePolicy à vos entités IAM. Cette politique est associée à un rôle lié à un service qui permet d' ElastiCache effectuer des actions en votre nom.

Cette politique permet ElastiCache de gérer les AWS ressources en votre nom selon les besoins de gestion de votre cache :

- `ec2`— Gérez les ressources EC2 réseau à associer aux nœuds de cache, notamment les points de terminaison VPC (pour les caches sans serveur), les interfaces réseau élastiques () (pour les clusters conçus par vos ENIs soins) et les groupes de sécurité.
- `cloudwatch`— Émet des données métriques depuis le service vers CloudWatch.
- `outposts`— Autorise la création de nœuds de cache sur les AWS Outposts.

Vous pouvez trouver la ElastiCacheServiceRolePolicypolitique sur la console IAM et ElastiCacheServiceRolePolicydans le AWS Managed Policy Reference Guide.

AWS politique gérée : AmazonElastiCacheFullAccess

Vous pouvez associer la politique `AmazonElastiCacheFullAccess` à vos identités IAM.

Cette politique donne aux principaux un accès complet à l' ElastiCache utilisation de la console AWS de gestion :

- `elasticache`— Accédez à tout APIs.
- `iam` : crée le rôle lié au service nécessaire au fonctionnement du service.
- `ec2`— Décrivez les EC2 ressources dépendantes nécessaires à la création du cache (VPC, sous-réseau, groupe de sécurité) et autorisez la création de points de terminaison VPC (pour les caches sans serveur).
- `kms`— Autoriser l'utilisation de services gérés par le client CMKs pour. encryption-at-rest
- `cloudwatch`— Autorisez l'accès aux métriques pour afficher ElastiCache les métriques dans la console.
- `application-autoscaling` : autorise l'accès pour décrire les politiques d'autoscaling des caches.
- `logs` : permet de remplir les flux de journaux pour la fonctionnalité de livraison des journaux dans la console.
- `firehose` : permet de remplir les flux de livraison pour la fonctionnalité de livraison des journaux dans la console.
- `s3` : permet de remplir les compartiments S3 pour la fonctionnalité de restauration des instantanés dans la console.
- `outposts`— Utilisé pour remplir les AWS Outposts afin de créer des caches dans la console.
- `sns` : permet de remplir les rubriques SNS pour la fonctionnalité de notification dans la console.

Vous pouvez trouver la AmazonElastiCacheFullAccesspolitique sur la console IAM et AmazonElastiCacheFullAccessdans le AWS Managed Policy Reference Guide.

AWS politique gérée : AmazonElastiCacheReadOnlyAccess

Vous pouvez associer la politique `AmazonElastiCacheReadOnlyAccess` à vos identités IAM.

Cette politique permet aux principaux d'accéder en lecture seule à la ElastiCache AWS console de gestion :

- `elasticache`— Accès en lecture seule `Describe` APIs.

Vous pouvez trouver la AmazonElastiCacheReadOnlyAccesspolitique sur la console IAM et AmazonElastiCacheReadOnlyAccessdans le AWS Managed Policy Reference Guide.

ElastiCache mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées ElastiCache 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 Historique du ElastiCache document.

Modification	Description	Date
AmazonElastiCacheFullAccess – Mise à jour d'une stratégie existante	ElastiCache a ajouté de nouvelles autorisations pour permettre la mise à l'échelle verticale pour MemCached , pour l'action`elasticache:ModifyCacheCluster` .	27 mars 2025
AmazonElastiCacheFullAccess – Mise à jour d'une politique existante	ElastiCache a ajouté de nouvelles autorisations pour permettre la gestion des caches sans serveur et pour permettre l'utilisation de toutes les fonctionnalités du service via la console.	27 novembre 2023
ElastiCacheServiceRolePolicy – Mise à jour d'une politique existante	ElastiCache a ajouté de nouvelles autorisations pour permettre la gestion des points de terminaison VPC pour les	27 novembre 2023

Modification	Description	Date
	ressources de cache sans serveur.	
ElastiCache a commencé à suivre les modifications	ElastiCache a commencé à suivre les modifications apportées AWS à ses politique s gérées.	7 février 2020

Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon ElastiCache

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 ElastiCache ressources Amazon. Pour de plus amples informations, veuillez consulter Vue d'ensemble de la gestion des autorisations d'accès à vos ElastiCache ressources.

Les sections de cette rubrique couvrent les sujets suivants :

- AWS politiques gérées pour Amazon ElastiCache
- Exemples de politiques gérées par le client

Voici un exemple de politique d'autorisation lors de l'utilisation de Redis OSS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowClusterPermissions",
```

```
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache",
                "elasticache:CreateCacheCluster",
                "elasticache:DescribeServerlessCaches",
                "elasticache:DescribeReplicationGroups",
                "elasticache:DescribeCacheClusters",
                "elasticache:ModifyServerlessCache",
                "elasticache:ModifyReplicationGroup",
                "elasticache:ModifyCacheCluster"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUserToPassRole",
            "Effect": "Allow",
            "Action": [ "iam:PassRole" ],
            "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
        }
    ]
}
```

Voici un exemple de politique d'autorisation lors de l'utilisation de Memcached.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowClusterPermissions",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache",
            "elasticache:CreateCacheCluster",
            "elasticache:DescribeServerlessCaches",
            "elasticache:DescribeCacheClusters",
            "elasticache:ModifyServerlessCache",
            "elasticache:ModifyCacheCluster"
        ],
        "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 déclaration accorde des autorisations pour les ElastiCache actions Amazon (`elasticache:Create*`,`elasticache:Describe*`,`elasticache:Modify*`)
- 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 consulter un tableau présentant toutes les actions de ElastiCache l'API Amazon et les ressources auxquelles elles s'appliquent, consultezElastiCache Autorisations d'API : référence aux actions, aux ressources et aux conditions.

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 ElastiCache lié à un service.

Associés aux autorisations minimales requises pour utiliser la ElastiCache console Amazon, les exemples de politiques présentés dans cette section accordent des autorisations supplémentaires. Les exemples sont également pertinents pour le AWS SDKs et le AWS CLI.

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 ElastiCache actions qui semblent simples peuvent nécessiter d'autres actions pour les prendre en charge lorsque vous utilisez la ElastiCache console. Par exemple, `elasticache:CreateCacheCluster` donne des autorisations pour créer des clusters de cache ElastiCache . Toutefois, pour effectuer cette opération, la ElastiCache console utilise un certain nombre d'`List`actions `Describe` et pour remplir les listes de consoles.

Exemples

- Exemple 1 : autoriser un utilisateur à accéder aux ressources en lecture seule ElastiCache
- Exemple 2 : autoriser un utilisateur à effectuer des tâches ElastiCache d'administrateur système courantes
- Exemple 3 : Autoriser un utilisateur à accéder à toutes les actions de ElastiCache l'API
- Exemple 4 : Autoriser un utilisateur à appeler l'API IAM CreateServiceLinkedRole
- Exemple 5 : Autoriser un utilisateur à se connecter à un cache sans serveur à l'aide de l'authentification IAM

Exemple 1 : autoriser un utilisateur à accéder aux ressources en lecture seule ElastiCache

La politique suivante accorde des autorisations à ElastiCache des actions qui permettent à un utilisateur de répertorier des ressources. En général, vous attachez ce type de politique d'autorisations à un groupe de gestionnaires.

JSON

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "ECReadOnly",
      "Effect":"Allow",
      "Action": [
          "elasticache:Describe*",
          "elasticache:List*"],
      "Resource":"*"
      }
```

```
   ]
}
```

Exemple 2 : autoriser un utilisateur à effectuer des tâches ElastiCache d'administrateur système courantes

Les tâches courantes d'administrateur système incluent la modification des ressources. Un administrateur système peut également obtenir des informations sur les événements ElastiCache . La politique suivante accorde à un utilisateur l'autorisation d'effectuer des ElastiCache actions relatives à ces tâches courantes d'administrateur système. Généralement, vous attachez ce type de politique d'autorisations au groupe d'administrateurs système.

JSON

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "ECAllowMutations",
      "Effect":"Allow",
      "Action":[
          "elasticache:Modify*",
          "elasticache:Describe*",
          "elasticache:ResetCacheParameterGroup"
      ],
      "Resource":"*"
      }
   ]
}
```

Exemple 3 : Autoriser un utilisateur à accéder à toutes les actions de ElastiCache l'API

La politique suivante permet à un utilisateur d'accéder à toutes les ElastiCache actions. Nous vous conseillons d'accorder ce type de politique d'autorisations uniquement à un utilisateur administrateur.

JSON

```
{
   "Version": "2012-10-17",
```

```
   "Statement":[{
      "Sid": "ECAllowAll",
      "Effect":"Allow",
      "Action":[
          "elasticache:*"
      ],
      "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. ElastiCache

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"CreateSLRAllows",
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringLike":{
          "iam:AWS ServiceName":"elasticache.amazonaws.com"
        }
      }
    }
  ]
}
```

Exemple 5 : Autoriser un utilisateur à se connecter à un cache sans serveur à l'aide de l'authentification IAM

La politique suivante permet à n'importe quel utilisateur de se connecter à n'importe quel cache sans serveur à l'aide de l'authentification IAM entre le 01/04/2023 et le 30/06/2023.

JSON

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
      }
    },
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:user:*"
      ]
    }
  ]
}
```

Autorisations de niveau ressource

Vous pouvez restreindre la portée des autorisations en spécifiant des ressources dans une politique IAM. De nombreuses actions d' ElastiCache 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.

Pour consulter la liste des types de ElastiCache ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par Amazon ElastiCache dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon ElastiCache.

Exemples

- Exemple 1 : accorder à un utilisateur un accès complet à des types de ElastiCache ressources spécifiques
- Exemple 2 : Refuser à un utilisateur l'accès à un cache sans serveur.

Exemple 1 : accorder à un utilisateur un accès complet à des types de ElastiCache ressources spécifiques

La politique suivante autorise explicitement toutes les ressources de type cache sans serveur.

```
{
        "Sid": "Example1",
        "Effect": "Allow",
        "Action": "elasticache:*",
        "Resource": [
             "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
        ]
}
```

Exemple 2 : Refuser à un utilisateur l'accès à un cache sans serveur.

L'exemple suivant refuse explicitement l'accès à un cache sans serveur particulier.

```
{
        "Sid": "Example2",
        "Effect": "Deny",
        "Action": "elasticache:*",
        "Resource": [
            "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
```

```
        ]
}
```

Utilisation de clés de condition

Vous pouvez spécifier des conditions pour déterminer comment une politique IAM prend effet. Dans ElastiCache, 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 ElastiCache condition, consultez la section Clés de condition pour Amazon ElastiCache 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.

Utilisation ElastiCache avec des clés de condition AWS globales

Lorsque vous utilisez des clés de condition AWS globales qui nécessitent ElastiCache le principal, utilisez une `OR` condition avec les deux principaux : `elasticache.amazonaws.com` et`ec.amazonaws.com`.

> **Note**
>
> Si vous n'ajoutez pas les deux principes pour ElastiCache, l'action « Autoriser » ou « Refuser » prévue ne sera appliquée correctement pour aucune des ressources répertoriées dans votre politique.

Exemple de politique avec clé de condition `aws:CalledVia` globale :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
```

```
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:CalledVia": [
            "ec.amazonaws.com",
            "elasticache.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

Spécification de conditions : Utilisation de clés de condition

Pour mettre en place un contrôle détaillé, vous écrivez une politique d'autorisations IAM qui spécifie les conditions pour contrôler un ensemble de paramètres individuels sur certaines demandes. Vous appliquez 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. Dans l'exemple suivant, vous spécifiez la condition selon laquelle tout cluster de cache auto-conçu créé sera de type nœud `cache.r5.large`.

> **Note**
>
> - Pour construire `Condition` des éléments à l'aide de clés de condition de `String` type, utilisez les opérateurs de condition qui ne distinguent pas les majuscules des `StringNotEqualsIgnoreCase` minuscules `StringEqualsIgnoreCase` ou pour comparer une clé à une valeur de chaîne.
> - ElastiCache traite les arguments d'entrée pour `CacheNodeType` et `CacheParameterGroupName` en faisant la distinction majuscules/minuscules. Pour cette raison, la chaîne conditionne les opérateurs `StringEqualsIgnoreCase` et `StringNotEqualsIgnoreCase` doit être utilisée dans les politiques d'autorisation qui les référencent.

Voici un exemple de cette politique d'autorisation lors de l'utilisation de Valkey ou Redis OSS.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.r5.large"
                    ]
                }
            }
        }
    ]
}
```

Voici un exemple de cette politique d'autorisation lors de l'utilisation de Memcached.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.r5.large"
                    ]
                }
            }
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter Marquer vos ressources ElastiCache .

Pour plus d'informations sur l'utilisation d'opérateurs de condition de politique, veuillez consulter ElastiCache Autorisations d'API : 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 permettant de mettre en œuvre un contrôle d'accès précis sur les paramètres listés ElastiCache précédemment.

1. elasticache : MaximumDataStorage : Spécifiez le stockage de données maximal d'un cache sans serveur. En utilisant les conditions fournies, le client ne peut pas créer de caches pouvant stocker une quantité de données supérieure à la quantité spécifiée.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
                "arn:aws:elasticache:*:*:snapshot:*",
                "arn:aws:elasticache:*:*:usergroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscache:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:MaximumDataStorage": "30"
                },
                "StringEquals": {
                    "elasticache:DataStorageUnit": "GB"
                }
            }
        }
```

```
    ]
}
```

2. ElastiCache:Maximum ECPUPer Second : Spécifiez la valeur maximale d'ECPU par seconde d'un cache sans serveur. En utilisant les conditions fournies, le client ne peut pas créer de caches capables d'exécuter plus d'un certain nombre de fois ECPUs par seconde.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
                "arn:aws:elasticache:*:*:snapshot:*",
                "arn:aws:elasticache:*:*:usergroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscache:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:MaximumECPUPerSecond": "100000"
                }
            }
        }
    ]
}
```

3. elasticache : CacheNodeType : Spécifiez ce NodeType qu'un utilisateur peut créer. En utilisant les conditions fournies, le client peut spécifier une valeur unique ou une plage pour un type de nœud.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.t2.micro",
                        "cache.t2.medium"
                    ]
                }
            }
        }
    ]
}
```

4. elasticache : CacheNodeType : Avec Memcached, spécifiez ce qu'un NodeType utilisateur peut créer. En utilisant les conditions fournies, le client peut spécifier une valeur unique ou une plage pour un type de nœud.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheNodeType": [
                        "cache.t2.micro",
                        "cache.t2.medium"
                    ]
                }
            }
        }
    ]
}
```

5. elasticache : NumNodeGroups : Créez un groupe de réplication avec moins de 20 groupes de nœuds.

   JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
             "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
             "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "NumericLessThanEquals": {
                    "elasticache:NumNodeGroups": "20"
                }
            }
        }
    ]
}
```

6. elasticache : ReplicasPerNodeGroup : Spécifiez les répliques par nœud entre 5 et 10.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
```

```
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "NumericGreaterThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "5"
                },
                "NumericLessThanEquals": {
                    "elasticache:ReplicasPerNodeGroup": "10"
                }
            }
        }
    ]
}
```

7. elasticache : EngineVersion : Spécifie l'utilisation de la version 5.0.6 du moteur.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
           "Effect": "Allow",
```

```
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:EngineVersion": "5.0.6"
                }
            }
        }
    ]
}
```

8. elasticache : EngineVersion : Spécifie l'utilisation de la version 1.6.6 du moteur Memcached

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "elasticache:EngineVersion": "1.6.6"
                }
            }
        }
    ]
}
```

9. elasticache : EngineType : Spécifiez en utilisant un moteur Valkey ou Redis OSS uniquement.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:EngineType": "redis"
                }
            }
        }
    ]
```

```
}
```

10. elasticache : AtRestEncryptionEnabled : Spécifiez que les groupes de réplication seront créés uniquement avec le chiffrement activé.

    JSON

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

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AtRestEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

11. Cache élastique : TransitEncryptionEnabled

    a. Définissez la clé de `elasticache:TransitEncryptionEnabled` condition sur `false` pour que l'CreateReplicationGroupaction spécifie que les groupes de réplication ne peuvent être créés que lorsque le protocole TLS n'est pas utilisé :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:TransitEncryptionEnabled": "false"
                }
            }
        }
    ]
}
```

Lorsque la clé de `elasticache:TransitEncryptionEnabled` condition est définie sur `false` dans une politique pour l'CreateReplicationGroupaction, une `CreateReplicationGroup` demande n'est autorisée que si le protocole TLS n'est pas utilisé (c'est-à-dire si la demande n'inclut aucun `TransitEncryptionEnabled` paramètre défini sur `true` ou défini sur). `TransitEncryptionMode required`

b. Définissez la clé de `elasticache:TransitEncryptionEnabled` condition sur `true` pour que l'CreateReplicationGroupaction spécifie que les groupes de réplication ne peuvent être créés que lorsque le protocole TLS est utilisé :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:TransitEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

Lorsque la clé de `elasticache:TransitEncryptionEnabled` condition est définie sur `true` dans une politique pour l'CreateReplicationGroupaction, une `CreateReplicationGroup` demande n'est autorisée que si la demande inclut un `TransitEncryptionEnabled` paramètre défini sur `true` et un `TransitEncryptionMode` paramètre défini sur`required`.

c. Définissez `elasticache:TransitEncryptionEnabled` sur `true` pour l'action `ModifyReplicationGroup` pour spécifier que les groupes de réplication ne peuvent être modifiés que lorsque le protocole TLS est utilisé :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:ModifyReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "BoolIfExists": {
                    "elasticache:TransitEncryptionEnabled": "true"
                }
            }
        }
    ]
}
```

Lorsque la clé de `elasticache:TransitEncryptionEnabled` condition est définie sur `true` dans une politique pour l'ModifyReplicationGroupaction, une `ModifyReplicationGroup` demande n'est autorisée que si elle inclut un `TransitEncryptionMode` paramètre défini sur`required`. Le paramètre `TransitEncryptionEnabled` défini sur `true` peut également être inclus en option, mais n'est pas nécessaire dans ce cas pour activer le protocole TLS.

12. elasticache : AutomaticFailoverEnabled : Spécifiez que les groupes de réplication seront créés uniquement lorsque le basculement automatique sera activé.

JSON

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

```
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AutomaticFailoverEnabled": "true"
                }
            }
        }
    ]
}
```

13 ElastiCache:Multi AZEnabled : Spécifiez que les groupes de réplication ne peuvent pas être créés lorsque le mode multi-AZ est désactivé.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Deny",
```

```
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:MultiAZEnabled": "false"
                }
            }
        }
    ]
}
```

14. elasticache : ClusterModeEnabled : Spécifiez que les groupes de réplication ne peuvent être créés qu'avec le mode cluster activé.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
```

```
            "Condition": {
                "Bool": {
                    "elasticache:ClusterModeEnabled": "true"
                }
            }
        }
    ]
}
```

15. elasticache : AuthTokenEnabled : Spécifiez que les groupes de réplication ne peuvent être créés qu'avec le jeton AUTH activé.

JSON

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

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "Bool": {
                    "elasticache:AuthTokenEnabled": "true"
                }
```

```
            }
        }
    ]
}
```

16. elasticache : SnapshotRetentionLimit : Spécifiez le nombre de jours (ou min/max) pour conserver le cliché. La politique ci-dessous applique le stockage des sauvegardes pendant au moins 30 jours.

    JSON

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

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup",
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*",
                "arn:aws:elasticache:*:*:serverlesscache:*"
            ],
            "Condition": {
                "NumericGreaterThanEquals": {
                    "elasticache:SnapshotRetentionLimit": "30"
                }
            }
```

```
        }
    ]
}
```

17. elasticache : KmsKeyId : Spécifiez l'utilisation des clés AWS KMS gérées par le client.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Sid": "AllowDependentResources",
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
            "arn:aws:elasticache:*:*:snapshot:*",
            "arn:aws:elasticache:*:*:usergroup:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticache:CreateServerlessCache"
        ],
        "Resource": [
            "arn:aws:elasticache:*:*:serverlesscache:*"
        ],
        "Condition": {
            "StringEquals": {
                "elasticache:KmsKeyId": "my-key"
            }
        }
    }
  ]
}
```

18. elasticache : CacheParameterGroupName : Spécifiez un groupe de paramètres autre que celui par défaut avec des paramètres spécifiques d'une organisation sur vos clusters. Vous pouvez également spécifier un modèle de dénomination pour vos groupes de paramètres ou supprimer

un bloc sur un nom de groupe de paramètres spécifique. Voici un exemple limitant l'utilisation de « my-org-param-group » uniquement.

JSON

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

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheParameterGroupName": "my-org-param-group"
                }
            }
        }
    ]
}
```

19. elasticache : CacheParameterGroupName : Avec Memcached, spécifiez un groupe de paramètres autre que celui par défaut avec des paramètres spécifiques d'une organisation sur vos clusters. Vous pouvez également spécifier un modèle de dénomination pour vos groupes de paramètres

ou supprimer un bloc sur un nom de groupe de paramètres spécifique. Voici un exemple limitant l'utilisation de « my-org-param-group » uniquement.

JSON

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

         {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },

        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "StringEquals": {
                    "elasticache:CacheParameterGroupName": "my-org-param-group"
                }
            }
        }
    ]
}
```

20. elasticache : CreateCacheCluster : Refus d'`CreateCacheCluster`action si la balise de requête `Project` est absente ou n'est pas égale à`Dev`, `QA` ou. `Prod`

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
          {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/Project": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster",
                "elasticache:AddTagsToResource"
            ],
            "Resource": "arn:aws:elasticache:*:*:cluster:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Project": [
                        "Dev",
                        "Prod",
                        "QA"
                    ]
                }
            }
```

```
        }
    ]
}
```

21. elasticache : CacheNodeType : Autoriser `CreateCacheCluster` avec `cacheNodeType` cache.r5.large ou cache.r6g.4xlarge et tag. `Project=XYZ`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
          ]
        },
        "StringEquals": {
          "aws:RequestTag/Project": "XYZ"
        }
      }
    }
  ]
```

```
}
```

22. elasticache : CacheNodeType : Autoriser `CreateCacheCluster` avec `cacheNodeType` cache.r5.large ou cache.r6g.4xlarge et tag. `Project=XYZ`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEqualsIfExists": {
          "elasticache:CacheNodeType": [
            "cache.r5.large",
            "cache.r6g.4xlarge"
          ]
        },
        "StringEquals": {
          "aws:RequestTag/Project": "XYZ"
        }
      }
    }
  ]
}
```

> **Note**
>
> Lors de la création de politiques pour appliquer ensemble des balises et d'autres clés de condition, la valeur `IfExists` conditionnelle peut être nécessaire sur des éléments de clés de condition en raison des exigences de politique `elasticache:AddTagsToResource` pour les demandes de création avec le paramètre `--tags`.

Utilisation de rôles liés à un service pour Amazon ElastiCache

Amazon ElastiCache utilise des rôles AWS Identity and Access Management liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à un AWS service, tel qu'Amazon. ElastiCache Les rôles ElastiCache liés aux services Amazon sont prédéfinis par Amazon. ElastiCache 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 d'Amazon ElastiCache , car vous n'avez pas à ajouter manuellement les autorisations nécessaires. Les rôles existent déjà dans votre AWS compte, mais ils sont liés à des cas ElastiCache d'utilisation Amazon et disposent d'autorisations prédéfinies. Seul Amazon ElastiCache 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 ElastiCache ressources Amazon, 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 Amazon ElastiCache
  - Autorisations pour créer un rôle lié à un service
- 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 Amazon ElastiCache
  - 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 Amazon ElastiCache
  - 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 Amazon ElastiCache

Autorisations pour créer un rôle lié à un service

Pour autoriser une entité IAM à créer un rôle lié à un AWS ServiceRoleForElastiCache 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/elasticache.amazonaws.com/AWS
 ServiceRoleForElastiCache*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

Pour autoriser une entité IAM à supprimer un rôle lié à un AWS ServiceRoleForElastiCache 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/elasticache.amazonaws.com/AWS
 ServiceRoleForElastiCache*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

Vous pouvez également utiliser une politique AWS gérée pour fournir un accès complet à Amazon ElastiCache.

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 https://console.aws.amazon.com/iam/l'adresse.
2. Dans le panneau de navigation de la console IAM, sélectionnez Roles (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 ElastiCache.
5. Choisissez Suivant : Autorisations.
6. Sous Policy name (Nom de la politique), notez que la `ElastiCacheServiceRolePolicy` 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 elasticache.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 `elasticache.amazonaws.com`.

Modification de la description d'un rôle lié à un service pour Amazon ElastiCache

Amazon ElastiCache ne vous autorise pas à modifier le rôle AWS ServiceRoleForElastiCache lié au 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 panneau de navigation de la console IAM, sélectionnez Roles (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 AWS ServiceRoleForElastiCache
   ```

   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 AWS ServiceRoleForElastiCache \
       --description "new description"
   ```

   Pour Windows :

   ```
   $ aws iam update-role-description ^
       --role-name AWS ServiceRoleForElastiCache ^
       --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=AWS ServiceRoleForElastiCache
   &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=AWS ServiceRoleForElastiCache
   &Version=2010-05-08
   &Description="New description"
```

Supprimer un rôle lié à un service pour Amazon ElastiCache

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.

Amazon ElastiCache ne supprime pas pour vous le rôle lié au service.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, confirmez d'abord qu'aucune session n'a de ressources (clusters ou groupes de réplication) associées 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 https://console.aws.amazon.com/iam/l'adresse.
2. Dans le panneau de navigation de la console IAM, sélectionnez Roles (Rôles). Choisissez ensuite le nom (et non la case à cocher) du AWS ServiceRoleForElastiCache 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 ElastiCache ressources Amazon qui nécessitent AWS ServiceRoleForElastiCache

- Pour supprimer un cluster, consultez les rubriques suivantes :
  - En utilisant le AWS Management Console
  - Utilisation du AWS CLI pour supprimer un ElastiCache cluster
  - Utilisation de l' ElastiCache API
- Pour supprimer un groupe de réplication, consultez les rubriques suivantes :
  - Suppression d'un groupe de réplication (console)
  - Suppression d'un groupe de réplication (AWS CLI)
  - Supprimer un groupe de réplication (ElastiCache API)

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 https://console.aws.amazon.com/iam/l'adresse.
2. Dans le panneau de navigation de la console IAM, sélectionnez Roles (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. Dans la boîte de dialogue de confirmation, passez en revue les données du dernier accès au service, qui indiquent la date à laquelle chacun des rôles sélectionnés a accédé à un AWS service pour la dernière fois. Cela vous permet de confirmer si le rôle est actif actuellement. 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 de ressources Amazon (ARNs) 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.

ElastiCache Autorisations d'API : 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 d' ElastiCache API Amazon 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 (*).

Vous pouvez utiliser des clés de condition dans vos ElastiCache politiques pour exprimer des conditions. Pour consulter la liste des clés de condition ElastiCache spécifiques, ainsi que les actions et les types de ressources auxquels elles s'appliquent, consultezUtilisation de clés de condition. Pour obtenir la liste complète des clés AWS-wide, voir les clés contextuelles de condition AWS globale dans le guide de l'utilisateur IAM.

> **Note**
>
> Pour indiquer une action, utilisez le préfixe `elasticache:` suivi du nom de l'opération d'API (par exemple, `elasticache:DescribeCacheClusters`).

Pour consulter la liste des ElastiCache actions, consultez la section Actions définies par Amazon ElastiCache dans le Service Authorization Reference.

Validation de conformité pour Amazon ElastiCache

Des auditeurs tiers évaluent la sécurité et la conformité des AWS services dans le cadre de plusieurs programmes de AWS conformité, tels que SOC, PCI, FedRAMP et HIPAA.

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

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 Services AWS 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 :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Référence des services éligibles HIPAA : liste les services éligibles HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- 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.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

- Lorsqu'Amazon ElastiCache est répertorié comme relevant du champ d'application d'un programme de conformité, ce champ d'application s'applique à tous les moteurs de cache pris en charge par ElastiCache. Cela inclut ElastiCache Valkey, ElastiCache Memcached et ElastiCache Redis OSS.

En savoir plus

Pour obtenir des informations générales sur la conformité au AWS cloud, consultez les rubriques suivantes :

- Points de terminaison FIPS par service
- Mises à jour du service dans ElastiCache
- AWS Conformité au cloud
- Modèle de responsabilité partagée
- AWS Programme de conformité à la norme PCI DSS

Résilience chez Amazon ElastiCache

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 mondiale, Amazon ElastiCache propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Rubriques

- Atténuation des défaillances

Atténuation des défaillances

Lorsque vous planifiez votre ElastiCache implémentation Amazon, 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.

Rubriques

- Atténuation des défaillances avec Memcached
- Atténuer les défaillances lors de l'exécution de Valkey ou Redis OSS
- Recommandations

Atténuation des défaillances avec Memcached

Lorsque vous exécutez le moteur Memcached, vous pouvez réduire au minimum l'impact d'un échec à l'aide des options suivantes. Il existe deux types de défaillances à prendre en compte dans vos plans d'atténuation des risques d'échec : les défaillances de nœud et les défaillances de zone de disponibilité.

Atténuation des défaillances de nœuds

Les caches sans serveur atténuent automatiquement les défaillances des nœuds grâce à une architecture multi-AZ répliquée de sorte que les défaillances des nœuds soient transparentes pour votre application. Afin d'atténuer l'impact d'une défaillance de nœud dans un cluster auto-conçu, répartissez vos données mises en cache sur plusieurs nœuds. Étant donné que les clusters auto-conçus ne prennent pas en charge la réplication, une défaillance de nœud entraînera toujours une perte de données de votre cluster.

Lorsque vous créez votre cluster Memcached, vous pouvez le créer avec 1 à 60 nœuds, ou plus sur demande spéciale. Le partitionnement de vos données sur un plus grand nombre de nœuds signifie que vous perdrez moins de données en cas de défaillance d'un nœud. Par exemple, si vous partitionnez vos données sur 10 nœuds, n'importe quel nœud simple stocke environ 10 % de vos données mises en cache. Dans ce cas, une défaillance de nœud perd environ 10 % de votre cache qui doit être remplacé lorsque le nœud de remplacement est créé et mis en service. Si les mêmes données ont été mis en cache dans 3 nœuds plus grands, la défaillance d'un nœud serait perdrait environ 33 % de vos données mises en cache.

Pour plus d'informations sur la spécification du nombre de nœuds dans un cluster Memcached, consultez Création d'un cluster Memcached (console).

Atténuation des défaillances des zones de disponibilité

Les caches sans serveur atténuent automatiquement les défaillances des zones de disponibilité grâce à une architecture multi-AZ répliquée de sorte que les défaillances des zones de disponibilité soient transparentes pour votre application.

Afin d'atténuer l'impact d'une défaillance de zone de disponibilité dans un cluster auto-conçu, recherchez vos nœuds dans le plus grand nombre possible de zones de disponibilité. Dans le cas peu probable d'une défaillance de la zone Z, vous perdrez les données mises en cache dans cette zone, et non les données mises en cache dans l'autre. AZs

Pourquoi un si grand nombre de nœuds ?

Si ma région a seulement 3 zones de disponibilité, pourquoi ai-je besoin de plus de 3 nœuds si je perds environ un tiers de mes données en cas de défaillance des zones de disponibilité ?

C'est une excellente question. Gardez à l'esprit que nous essayons d'atténuer deux types de défaillances distincts, au niveau du nœud et au niveau de la zone de disponibilité. Vous avez raison, si vos données sont réparties dans les zones de disponibilité et que l'une des zones échoue, vous perdrez uniquement les données mises en cache de cette zone, quel que soit votre nombre de nœuds. Toutefois, si un nœud échoue, le fait d'avoir plusieurs nœuds permet de réduire la proportion des données mises en perdues.

Il n'y a pas de « formule magique » pour déterminer le nombre de nœuds idéal qu'un cluster doit avoir. Vous devez mesurer l'impact de la perte de données par rapport à la probabilité d'une défaillance, par rapport au coût, et tirer vos propres conclusions.

Pour plus d'informations sur la spécification du nombre de nœuds dans un cluster Memcached, consultez Création d'un cluster Memcached (console).

Pour plus d'informations sur les régions et les zones de disponibilité, consultez Choix des régions et des zones de disponibilité pour ElastiCache.

Atténuer les défaillances lors de l'exécution de Valkey ou Redis OSS

Lorsque vous exécutez un moteur Valkey ou Redis OSS, vous disposez des options suivantes pour minimiser l'impact d'une défaillance d'un nœud ou d'une zone de disponibilité.

Atténuation des défaillances de nœuds

Les caches sans serveur atténuent automatiquement les défaillances des nœuds grâce à une architecture multi-AZ de sorte que les défaillances des nœuds soient transparentes pour votre application. Les clusters auto-conçus doivent être configurés de manière appropriée pour atténuer la défaillance d'un nœud individuel.

Pour atténuer l'impact des défaillances des nœuds Valkey ou Redis OSS sur les clusters conçus par vos soins, vous disposez des options suivantes :

Rubriques

- Atténuation des défaillances : groupes de réplication Valkey ou Redis OSS

Atténuation des défaillances : groupes de réplication Valkey ou Redis OSS

Un groupe de réplication Valkey ou Redis OSS est composé d'un seul nœud principal sur lequel votre application peut à la fois lire et écrire, et de 1 à 5 nœuds de réplication en lecture seule. Chaque fois que des données sont écrites sur le nœud principal, elles sont également mises à jour de façon asynchrone sur les nœuds de réplica en lecture.

Cas d'échec d'un réplica en lecture

1. ElastiCache détecte l'échec de lecture de la réplique.
2. ElastiCache met le nœud défaillant hors ligne.
3. ElastiCache lance et approvisionne un nœud de remplacement dans la même zone de disponibilité.
4. La nouveau nœud est synchronisé sur le nœud principal.

Pendant ce temps, votre application peut continuer à lire et à écrire en utilisant les autres nœuds.

Valkey ou Redis OSS Multi-AZ

Vous pouvez activer le Multi-AZ sur vos groupes de réplication Valkey ou Redis OSS. Que vous activiez Multi-AZ ou non, un nœud principal défaillant sera détecté et remplacé automatiquement. Ce qui se produit dépend du fait que Multi-AZ soit activé ou pas.

Lorsque Multi-AZ est activé

1. ElastiCache détecte la défaillance du nœud principal.

2. ElastiCache promeut le nœud de réplication en lecture avec le moins de retard de réplication vers le nœud principal.
3. La synchronisation des autres réplicas avec le nouveau nœud principal se produit.
4. ElastiCache lance une réplique en lecture dans l'AZ du serveur principal défaillant.
5. La synchronisation du nouveau nœud avec le réplica principal nouvellement promu se produit.

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 que si Multi-AZ n'avait pas été activé.

Pour de plus amples informations, veuillez consulter Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS.

Lorsque Multi-AZ est désactivé

1. ElastiCache détecte une défaillance principale.
2. ElastiCache met le serveur principal hors ligne.
3. ElastiCache crée et approvisionne un nouveau nœud principal pour remplacer le nœud principal défaillant.
4. ElastiCache synchronise le nouveau primaire avec l'une des répliques existantes.
5. Lorsque la synchronisation est terminée, le nouveau nœud fonctionne en tant que nœud principal du cluster.

Pendant les étapes 1 à 4 de ce processus, votre application ne peut pas écrire sur le nœud principal. Toutefois, votre application peut continuer la lecture à partir de vos nœuds de réplica.

Pour une protection accrue, nous vous recommandons de lancer les nœuds de votre groupe de réplication dans différentes zones de disponibilité (AZs). Si vous procédez ainsi, une défaillance de zone de disponibilité affectera uniquement les nœuds figurant dans cette zone et non ceux des autres zones.

Pour de plus amples informations, veuillez consulter Haute disponibilité avec les groupes de réplication.

Atténuation des défaillances des zones de disponibilité

Les caches sans serveur atténuent automatiquement les défaillances des zones de disponibilité grâce à une architecture multi-AZ répliquée de sorte que les défaillances des zones de disponibilité soient transparentes pour votre application.

Afin d'atténuer l'impact d'une défaillance de zone de disponibilité dans un cluster auto-conçu, recherchez vos nœuds pour chaque partition dans le plus grand nombre possible de zones de disponibilité.

Quel que soit le nombre de nœuds que contient une partition, s'ils sont tous situés dans la même zone de disponibilité, une défaillance catastrophique de la zone de disponibilité entraînera la perte de toutes les données de votre partition. Toutefois, si vous localisez vos nœuds dans plusieurs zones AZs, en cas de défaillance d'une zone de zone, vous ne perdez que les nœuds de cette zone de zone.

A chaque fois que vous perdez un nœud, vous notez une dégradation des performances puisque les opérations de lecture sont désormais partagées par un plus petit nombre de nœuds. Cette dégradation des performances perdurera tant le nœuds n'auront pas été remplacés.

Pour plus d'informations sur la spécification des zones de disponibilité pour les nœuds Valkey ou Redis OSS, consultez. Création d'un cluster Valkey (mode cluster désactivé) (console)

Pour plus d'informations sur les régions et les zones de disponibilité, consultez Choix des régions et des zones de disponibilité pour ElastiCache.

Recommandations

Nous vous recommandons de créer des caches sans serveur sur des clusters auto-conçus, car vous obtiendrez automatiquement une meilleure tolérance aux pannes sans configuration supplémentaire. Lors de la création d'un cluster auto-conçu, vous devez toutefois prévoir deux types de défaillances : les défaillances au niveau du nœud et les défaillances plus étendues au niveau des zones de disponibilité. Le meilleur plan d'atténuation des risques d'échec traitera deux types de défaillances.

Réduction au maximum de l'impact des défaillances des nœuds

Pour minimiser l'impact d'une panne de nœud lors de l'utilisation de Valkey ou Redis OSS, nous recommandons que votre implémentation utilise plusieurs nœuds dans chaque partition et répartisse les nœuds sur plusieurs zones de disponibilité. Ce processus est automatique pour les caches sans serveur.

Pour les clusters conçus par vos soins sur Valkey ou Redis OSS, nous vous recommandons d'activer le mode multi-AZ sur votre groupe de réplication afin qu'il bascule ElastiCache automatiquement vers une réplique en cas de défaillance du nœud principal.

Si Memcached est exécuté et que vos données sont partitionnées sur plusieurs nœuds, plus les nœuds que vous utilisez seront nombreux, moins la perte de données sera importante en cas de défaillance d'un des nœuds.

Minimiser l'impact des défaillances au niveau des zones de disponibilité

Afin de limiter l'impact d'une défaillance de zone de disponibilité, nous vous recommandons de lancer vos nœuds dans le plus grand nombre de zones de disponibilité disponibles. En répartissant vos nœuds de manière uniforme, AZs vous minimiserez l'impact dans le cas peu probable d'une défaillance de l'AZ. Ce processus est automatique pour les caches sans serveur.

Autres précautions

Si vous utilisez Valkey ou Redis OSS, en plus de ce qui précède, nous vous recommandons de planifier des sauvegardes régulières de votre cluster. Les sauvegardes (instantanés) créent un fichier .rdb, que vous pouvez utiliser pour restaurer votre cache en cas de défaillance ou d'endommagement. Pour de plus amples informations, veuillez consulter Instantané et restauration.

Sécurité de l'infrastructure dans AWS ElastiCache

En tant que service géré, AWS ElastiCache il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans la section Sécurité et conformité d'AWS Architecture Center.

Vous utilisez des appels d'API AWS publiés pour accéder ElastiCache 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.

Mises à jour du service dans ElastiCache

ElastiCache surveille automatiquement votre parc de caches, de clusters et de nœuds pour appliquer les mises à jour de service dès qu'elles sont disponibles. Les mises à jour de service des caches sans serveur sont appliquées automatiquement et de manière transparente. Pour les clusters conçus par vos soins, vous définissez une fenêtre de maintenance prédéfinie afin de ElastiCache pouvoir 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 à vos clusters auto-conçus et à quel moment. Vous pouvez également suivre la progression de ces mises à jour sur le ElastiCache cluster sélectionné en temps réel.

Rubriques

- Gestion des mises à jour de service pour les clusters conçus par vos soins

Gestion des mises à jour de service pour les clusters conçus par vos soins

ElastiCache des mises à jour de service pour les clusters auto-conçus sont publiées régulièrement. Si vous disposez d'un ou de plusieurs clusters conçus par vos soins pour ces mises à jour de service, vous recevez des notifications par e-mail, via le réseau de 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 Mises à jour des services de la ElastiCache console. En utilisant ce tableau de bord, vous pouvez consulter toutes les mises à jour du service et leur statut pour votre ElastiCache flotte. Les mises à jour de service des caches sans serveur sont appliquées de manière transparente et ne peuvent pas être gérées via les mises à jour de service.

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 mise à jour de sécurité dès que possible afin de garantir que vos ElastiCache clusters disposent toujours des correctifs de sécurité up-to-date les plus récents.

Les sections suivantes décrivent ces options en détail :

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 noter toute action lorsqu'elle sera disponible. ElastiCache planifiera d'appliquer la mise à jour lors de l'une des prochaines fenêtres 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 à un service aux ElastiCache clusters 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 ElastiCache clusters, vous pouvez utiliser la ElastiCache console, l'API ou AWS CLI 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 ElastiCache console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le panneau de navigation, choisissez Service Updates (Mises à jour des services).
3. Sous Service updates (Mises à jour de services), vous pouvez afficher les éléments suivants :
   - Nom de la mise à jour de service : le nom unique de la mise à jour de service
   - 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)
   - Gravité de la mise à jour : la priorité d'application de la mise à jour :
     - critical (critique) : nous vous recommandons d'effectuer cette mise à jour immédiatement (dans un délai de 14 jours maximum).

  - important (importante) : nous vous recommandons d'effectuer cette mise à jour dès que votre flux de travail vous le permettra (dans un délai de 30 jours maximum).
  - medium : nous vous recommandons d'effectuer cette mise à jour dès que possible (dans un délai de 60 jours maximum).
  - low (faible) : nous vous recommandons d'effectuer cette mise à jour dès que possible (dans un délai de 90 jours maximum).
- Version du moteur : si le type de mise à jour est engine-update, la version du moteur qui est mise à jour.
- Date de publication : lorsque la mise à jour est publiée et disponible pour application sur vos clusters.
- Recommended Apply By Date (Date limite d'application recommandée) : date de référence limite ElastiCache pour l'application des mises à jour.
- Status (État) : état de la mise à jour, à savoir l'un des états suivants :
  - available : la mise à jour est disponible pour les clusters requis.
  - achevée : la mise à jour a été effectuée.
  - cancelled (annulée) : la mise à jour a été annulée et n'est plus nécessaire.
  - expirée : la mise à jour n'est plus disponible.

4. Choisissez une mise à jour individuelle (et non le bouton à sa gauche) pour afficher les détails de la mise à jour de service.

   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.
     - in-progress : 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 elasticache describe-service-updates --service-update-status
  available
  ```

  Pour de plus amples informations, veuillez consulter describe-service-updates.
- Pour appliquer une mise à jour de service à une liste de clusters, exécutez la commande suivante :

  ```
  aws elasticache batch-apply-update-action --service-update
  ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
  cluster2
  ```

  Pour de plus amples informations, veuillez consulter batch-apply-update-action.

Vérification de l'application de la dernière mise à jour de service à l'aide de la AWS console

Vous pouvez vérifier que vos ElastiCache clusters Redis OSS exécutent la dernière mise à jour du service en procédant comme suit :

1. Choisissez un cluster applicable sur la page Redis OSS Clusters
2. Choisissez Mises à jour de service dans le volet de navigation pour voir les mises à jour de service applicables à ce cluster, le cas échéant.

Si la console affiche une liste des mises à jour de service, vous pouvez sélectionner la mise à jour de service et choisir Appliquer maintenant.

Si la console affiche « Aucune mise à jour de service trouvée », cela signifie que la dernière ElastiCache mise à jour de service est déjà appliquée au cluster Redis OSS.

Gestion des mises à jour du service

Si nécessaire, vous pouvez arrêter les mises à jour des clusters. Par exemple, vous pouvez le faire en cas d'augmentation inattendue du nombre de clusters faisant l'objet d'une mise à jour. Vous pouvez également choisir cette option si les mises à jour sont trop longues et qu'elles interrompent votre flux d'activité en période de pointe.

L'opération d'arrêt interrompt immédiatement toutes les mises à jour des clusters et de tous les nœuds qui doivent encore être mis à jour. Tous les nœuds dont l'état est in progress (en cours) sont achevés. Toutefois, il interrompt les mises à jour d'autres nœuds du même cluster dont l'état est update available (mise à jour disponible) et les repasse en état Stopping (Arrêt en cours).

Lorsque le workflow Stopping (Arrêt en cours est terminé, les nœuds dont l'état est Stopping (Arrêt en cours) passent à un état Stopped (Arrêté). Selon le workflow de la mise à jour, certains clusters n'auront pas de nœuds mis à jour. D'autres clusters peuvent inclure certains nœuds qui sont mis à jour et d'autres dont l'état est encore update available (mise à jour disponible).

Vous pouvez revenir plus tard pour terminer le processus de mise à jour dans la mesure où vos flux d'activité le permettent. Dans ce cas, sélectionnez les clusters applicables sur lesquels vous souhaitez effectuer les mises à jour, puis sélectionnez Apply now (Appliquer maintenant). Pour de plus amples informations, veuillez consulter Application des mises à jour de service.

Utilisation de la console

Vous pouvez interrompre la mise à jour d'un service à l'aide de la ElastiCache console. L'exemple suivant vous montre comment procéder :

- Une fois qu'une mise à jour de service a progressé sur un cluster sélectionné, la ElastiCache console affiche l'onglet Afficher/Arrêter la mise à jour en haut du tableau de bord. ElastiCache
- Pour interrompre la mise à jour, sélectionnez Stop Update (Arrêter la mise à jour).
- Lorsque vous arrêtez la mise à jour, choisissez le cluster et examinez son statut. Il revient d'abord à l'état Stopping (Arrêt en cours), puis Stopped (Arrêté).

À l'aide du AWS CLI

Vous pouvez interrompre une mise à jour du service à l'aide de la AWS CLI. L'exemple de code suivant montre comment procéder :

Pour un groupe de réplication, procédez comme suit :

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Pour un cluster de cache, procédez comme suit :

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

Pour de plus amples informations, veuillez consulter BatchStopUpdateAction.

Vulnérabilités et expositions courantes (CVE) : vulnérabilités de sécurité corrigées dans ElastiCache

Vulnérabilités et expositions courantes (CVE) est une liste d'entrées pour les vulnérabilités de cybersécurité connues du public. Chaque entrée est un lien qui contient un numéro d'identification,

une description et au moins une référence publique. Vous trouverez sur cette page une liste des failles de sécurité qui ont été corrigées dans ElastiCache.

Nous vous recommandons de toujours passer aux dernières versions de ElastiCache Valkey, Redis OSS ou ElastiCache Memcached afin de vous protéger contre les vulnérabilités connues. Lorsque vous utilisez un cache ElastiCache sans serveur, les correctifs CVE sont automatiquement appliqués à votre cache. Lorsque vous utilisez des clusters conçus par vos soins avec Valkey ou Redis OSS, le composant PATCH est ElastiCache exposé. Par exemple, lors de l'utilisation ElastiCache de la version 6.2.6 de Redis OSS, la version majeure est 6, la version mineure est 2 et la version du correctif est 6. Les versions PATCH sont destinées aux corrections de bogues rétrocompatibles, aux correctifs de sécurité et aux modifications non fonctionnelles.

Vous pouvez utiliser le tableau suivant pour vérifier si une version particulière de ElastiCache Valkey et Redis OSS contient un correctif pour une vulnérabilité de sécurité spécifique. Si votre cluster ElastiCache Valkey ou Redis OSS exécute une version sans le correctif de sécurité, reportez-vous au tableau ci-dessous et prenez les mesures nécessaires. Vous pouvez soit effectuer une mise à niveau vers une version plus récente de ElastiCache Valkey ou Redis OSS contenant le correctif, soit, si vous utilisez une version contenant le correctif, vous assurer que la dernière mise à jour du service est appliquée en vous référant à. Gestion des mises à jour de service pour les clusters conçus par vos soins Pour plus d'informations sur les versions de ElastiCache moteur prises en charge et sur la procédure de mise à niveau, consultezVersions du moteur et mise à niveau dans ElastiCache.

> **Note**
>
> - Si un CVE est traité dans une ElastiCache version, cela signifie qu'il est également traité dans les versions plus récentes. Ainsi, par exemple, si une vulnérabilité est corrigée dans ElastiCache la version 6.0.5 de Redis OSS, elle se poursuit dans les versions 6.2.6, 7.0.7 et 7.1.
> - Un astérisque (*) dans le tableau suivant indique que la dernière mise à jour de service doit être appliquée ElastiCache pour le cluster Redis OSS exécutant la version ElastiCache pour Redis OSS spécifiée afin de corriger la vulnérabilité de sécurité. Pour plus d'informations sur la façon de vérifier que la dernière mise à jour de service est appliquée ElastiCache pour la version Redis OSS sur laquelle votre cluster s'exécute, consultezGestion des mises à jour de service pour les clusters conçus par vos soins.

ElastiCache pour la version Redis OSS	CVEs Adressée
Redis OSS 6.0.5	CVE-24735 *, CVE-24736 *
Redis OSS 6.2.6	CVE-24834 *, CVE-35977 *, CVE-36021 *, CVE-24735, CVE-24736, CVE-2023-22458 , CVE-2023-25155 CVE-2023-28856, CVE-2023-45145
Redis OSS 7.0.7	CVE-2023-41056* , CVE-24834 *, CVE-35977, CVE-36021, CVE-24735, CVE-24736
Redis OSS 7.1.0	CVE-2023-41056, CVE-24834, CVE-35977 , CVE-42/36021, CVE-24735, CVE-24736

Journalisation et surveillance sur Amazon ElastiCache

Pour gérer votre cache, il est important que vous connaissiez ses performances. ElastiCache génère des métriques qui sont publiées sur Amazon CloudWatch Logs pour surveiller les performances de votre cache. En outre, ElastiCache génère des événements lorsque des modifications importantes se produisent sur les ressources de votre cache (par exemple, un nouveau cache est créé ou un cache est supprimé).

Rubriques

- Métriques et événements sans serveur pour Valkey et Redis OSS
- Mesures et événements de cluster conçus par vos soins pour Valkey et Redis OSS
- Métriques et événements pour Memcached
- Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail
- Surveillance des événements par Amazon SNS ElastiCache
- Livraison de journaux
- Surveillance de l'utilisation à l'aide de CloudWatch métriques

- Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail

Métriques et événements sans serveur pour Valkey et Redis OSS

ElastiCache propose un large éventail de mesures et d'événements à surveiller lorsque vous travaillez avec des caches sans serveur. Cela inclut CloudWatch les métriques, les métriques au niveau des commandes et les journaux d'événements qui peuvent être ingérés via Amazon EventBridge.

Rubriques

- Métriques pour les caches sans serveur
- Événements pour les caches sans serveur

Métriques pour les caches sans serveur

L'espace de `AWS/ElastiCache` noms inclut les CloudWatch métriques suivantes pour vos caches sans serveur Valkey ou Redis OSS.

Codes métriques pour Valkey ou Redis OSS

Métrique	Description	Unit
BytesUsedForCache	Nombre total d'octets utilisés par les données stockées dans votre cache.	Octets
ElastiCacheProcessingUnits	Le nombre total de ElastiCacheProcessingUnits (ECPUs) consommés par les requêtes exécutées sur votre cache	Nombre
SuccessfulReadRequestLatency	Latence des demandes de lecture réussies.	Microsecondes
SuccessfulWriteRequestLatency	Latence des demandes d'écriture réussies.	Microsecondes
TotalCmdsCount	Nombre total de commandes exécutées sur votre cache.	Nombre
CacheHitRate	Indique le taux d'accès à votre cache. Ce ratio est calculé en utilisant les statistiques `cache_hits` et `cache_misses` de la manière suivante : `cache_hits /(cache_hits + cache_misses)` .	Pourcentage
CacheHits	Nombre de recherches réussies de clés en lecture seule dans le cache.	Nombre
CurrConnections	Nombre de connexions client à votre cache.	Nombre
ThrottledCmds	Le nombre de demandes qui ont été limitées ElastiCache	Nombre

Métrique	Description	Unit
	parce que la charge de travail augmentait plus rapidement que ce que l' ElastiCache on pouvait faire.	
NewConnections	Nombre total de connexions qui ont été acceptées par le serveur au cours de cette période.	Nombre
CurrItems	Nombre d'éléments dans le cache.	Nombre
CurrVolatileItems	Nombre d'éléments dans le cache ayant une durée de vie.	Nombre
NetworkBytesIn	Nombre total d'octets transférés dans le cache.	Octets
NetworkBytesOut	Nombre total d'octets transférés hors du cache.	Octets
Evictions	Nombre de clés expulsées par le cache.	Nombre
IamAuthenticationExpirations	Le nombre total de connexions Valkey ou Redis OSS authentifiées par IAM expirées. Vous trouverez plus d'informations sur Authentification avec IAM dans le guide de l'utilisateur.	Nombre

Métrique	Description	Unit
IamAuthenticationThrottling	Le nombre total de demandes Valkey ou Redis OSS AUTH ou HELLO limitées authentif iées par IAM. Vous trouverez plus d'informations sur Authentification avec IAM dans le guide de l'utilisateur.	Nombre
KeyAuthorizationFailures	Nombre total de tentative s infructueuses par les utilisateurs d'accéder aux clés auxquelles ils n'ont pas l'autorisation d'accéder. Nous vous suggérons de déclenche r une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
AuthenticationFailures	Nombre total de tentatives infructueuses d'authentification auprès de Valkey ou Redis OSS à l'aide de la commande AUTH. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre

Métrique	Description	Unit
CommandAuthorizationFailures	Nombre total de tentatives infructueuses par les utilisateurs d'exécuter des commandes qu'ils n'ont pas l'autorisation d'appeler. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre

Métriques au niveau de la commande

ElastiCache émet également les métriques de niveau commande suivantes. Pour chaque type de commande, ElastiCache émet le nombre total de commandes et le nombre de commandes ECPUs consommées par ce type de commande.

Métrique	Description	Unit
EvalBasedCmds	Nombre de commandes get reçues par le cache.	Nombre
EvalBasedCmdsECPUs	ECPUs consommé par les commandes basées sur l'égalité.	Nombre
GeoSpatialBasedCmds	Nombre total de commandes pour les commandes basées sur la géolocalisation. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS. Il est dérivé en additionnant tous les types de commandes geo : geoadd, geodist, geohash, geopos,	Nombre

Métrique	Description	Unit
	georadius et georadius bymember.	
GeoSpatialBasedCmdsECPUs	ECPUs consommé par les commandes géospatiales.	Nombre
GetTypeCmds	Nombre total de commandes de type lecture seule. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes de type lecture seule (get, hget, scard, lrange, etc.)	Nombre
GetTypeCmdsECPUs	ECPUs consommé par les commandes de lecture.	Nombre
HashBasedCmds	Nombre total de commandes basées sur le hachage. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur un ou plusieurs hachages (hget, hkeys, hvals, hdel, etc.).	Nombre
HashBasedCmdsECPUs	ECPUs consommé par des commandes basées sur le hachage.	Nombre

Métrique	Description	Unit
HyperLogLogBasedCmds	Le nombre total de commandes HyperLogLog basées. Ceci est dérivé des statistiques de commande Valkey ou Redis OSS en additionnant tous les types de commandes pf (pfadd, pfcount, pfmerge, etc.).	Nombre
HyperLogLogBasedCmdsECPUs	ECPUs consommé par HyperLogLog des commandes basées.	Nombre
JsonBasedCmds	Nombre total de commandes JSON, y compris les commandes de lecture et d'écriture. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes JSON qui agissent sur les clés JSON.	Nombre
JsonBasedCmdsECPUs	ECPUs consommé par toutes les commandes JSON, y compris les commandes de lecture et d'écriture.	Nombre

Métrique	Description	Unit
JsonBasedGetCmds	Nombre total de commandes en lecture seule JSON. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes de lecture JSON qui agissent sur les clés JSON.	Nombre
JsonBasedGetCmdsECPUs	ECPUs consommé par les commandes JSON en lecture seule.	Nombre
JsonBasedSetCmds	Nombre total de commandes d'écriture JSON. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionn ant toutes les commandes d'écriture JSON qui agissent sur les clés JSON.	Nombre
JsonBasedSetCmdsECPUs	ECPUs consommé par les commandes d'écriture JSON.	Nombre
KeyBasedCmds	Nombre total de commandes basées sur une clé. Ceci est dérivé de la statistique des commandes Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur une ou plusieurs touches dans plusieurs structures de données (del, expire, rename, etc.).	Nombre

Métrique	Description	Unit
KeyBasedCmdsECPUs	ECPUs consommé par des commandes basées sur des touches.	Nombre
ListBasedCmds	Nombre total de commandes basées sur une liste. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur une ou plusieurs listes (lindex, lrange, lpush, ltrim, etc.).	Nombre
ListBasedCmdsECPUs	ECPUs consommé par des commandes basées sur des listes.	Nombre
NonKeyTypeCmds	Nombre total de commandes non basées sur des clés. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes qui n'agissent pas sur une touche, par exemple acl, dbsize ou info.	Nombre
NonKeyTypeCmdsECPUs	ECPUs consommé par non-key-based les commandes.	Nombre

Métrique	Description	Unit
PubSubBasedCmds	Nombre total de commandes pour les pub/sub fonctionn alités. Ceci est dérivé des statistiques de commande Valkey ou Redis OSS en additionnant toutes les commandes utilisées pour les pub/sub fonctionnalités : psubscribe, publish, pubsub, punsubscribe, ssubscrib e, sunsubscribe, spublish, subscribe et unsubscribe.	Nombre
PubSubBasedCmdsECPUs	ECPUs consommé par les commandes basées sur pub/sub.	Nombre
SetBasedCmds	Nombre total de commandes basées sur un ensemble. Ceci est dérivé de la statistique des commandstats de Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur un ou plusieurs ensembles (scard, sdiff, sadd, sunion, etc.).	Nombre
SetBasedCmdsECPUs	ECPUs consommé par des commandes basées sur des ensembles.	Nombre

Métrique	Description	Unit
SetTypeCmds	Nombre total de commandes de type écriture. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant tous les types mutatifs de commandes qui opèrent sur les données (set, hset, sadd, lpop, etc.)	Nombre
SetTypeCmdsECPUs	ECPUs consommé par les commandes d'écriture.	Nombre
SortedSetBasedCmds	Nombre total de commandes qui sont triées en fonction d'un ensemble. Ceci est dérivé de la statistique des commandes Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur un ou plusieurs ensembles triés (zcount, zrange, zrank, zadd, etc.).	Nombre
SortedSetBasedCmdsECPUs	ECPUs consommé par des commandes basées sur le tri.	Nombre

Métrique	Description	Unit
StringBasedCmds	Nombre total de commandes basées sur une chaîne. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionn ant toutes les commandes qui agissent sur une ou plusieurs chaînes (strlen, setex, setrange, etc.).	Nombre
StringBasedCmdsECPUs	ECPUs consommé par des commandes basées sur des chaînes.	Nombre
StreamBasedCmds	Nombre total de commandes basées sur un flux. Ceci est dérivé de la statistique commandstats de Valkey ou Redis OSS en additionnant toutes les commandes qui agissent sur un ou plusieurs types de données de flux (xrange, xlen, xadd, xdel, etc.).	Nombre
StreamBasedCmdsECPUs	ECPUs consommé par des commandes basées sur des flux.	Nombre

Événements pour les caches sans serveur

ElastiCache enregistre les événements liés à votre cache sans serveur. 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 ElastiCache console, de la commande AWS CLI describe-events ou de l'action ElastiCache API. `DescribeEvents`

Vous pouvez choisir de surveiller, d'ingérer, de transformer les ElastiCache événements et d'agir en conséquence à l'aide d'Amazon EventBridge. Pour en savoir plus, consultez le guide de EventBridge démarrage Amazon.

Affichage ElastiCache des événements (console)

Pour afficher les événements à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/
2. Dans le volet de navigation, choisissez Événements pour consulter une liste de tous les événements disponibles.
3. Sur l'écran Événements, chaque ligne de la liste représente un événement et affiche sa source, son type, son heure GMT et sa description. 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.

Afficher ElastiCache les événements (AWS CLI)

Pour générer une liste d' ElastiCache événements 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 cache sans serveur.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Le code suivant répertorie tous les événements de caches sans serveur qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Événements pour les caches sans serveur

Cette section décrit les différents types d'événements que vous pouvez recevoir pour vos caches sans serveur.

Événements de création de caches sans serveur

Detail-Type	Description	Unit	Source	Message
Cache créé	ARN du cache	création	serverless-cache	Le cache <nom-cache> est créé et prêt à être utilisé.
Cache créé	ARN du cache ARN de l'instantané	création	serverless-cache	Le cache <nom-cache> est créé et les données ont été restaurées à partir de l'instantané. Votre cache est prêt à être utilisé.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Nombre d'adresses IP libres insuffisant pour créer un point de terminaison de VPC.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Sous-réseaux non valides fournis dans la demande.

Detail-Type	Description	Unit	Source	Message
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Limite de quota atteinte pour la création d'un point de terminaison de VPC.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Vous n'êtes pas autorisé à créer un point de terminaison de VPC.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Un utilisateur dont la version de Valkey ou Redis OSS est incompatible est présent dans le groupe d'utilisa teurs < >user-gro up-name.

Detail-Type	Description	Unit	Source	Message
Échec de la création du cache	ARN du cache ARN de l'instantané du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Le groupe d'utilisateurs fourni <user-group-name> n'existe pas.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. La restauration des données à partir de l'instantané a échoué, car <motif>. Motifs de l'échec : • la récupération du fichier à partir de S3 a échoué. • le md5 attendu ne correspond pas au md5 réel. • le fichier RDB fourni possède une version non prise en charge.

Événements de mise à jour du cache sans serveur (Valkey ou Redis OSS)

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	SecurityGroups mis à jour pour le cache<cache-name>.
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	Balises mises à jour pour le cache <nom-cache>.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <nom-cache> a échoué. Un utilisateur dont la version de Valkey ou Redis OSS est incompatible est présent dans le groupe d'utilisateurs < >user-group-name.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-name>a échoué. SecurityGroups échec de la mise à jour.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-

Detail-Type	Liste de ressources	Catégorie	Source	Message
				name>a échoué. SecurityGroups la mise à jour a échoué en raison d'autorisations insuffisantes.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-name>a échoué. SecurityGroups la mise à jour a échoué car SecurityGroups ils ne sont pas valides.

Événements de suppression du cache sans serveur (Valkey ou Redis OSS)

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache supprimé	ARN du cache	suppression	serverless-cache	Le cache <nom-cache> a été supprimé.

Événements de limite d'utilisation du cache sans serveur (Valkey ou Redis OSS)

Detail-Type	Description	Unit	Source	Message
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	Limites mises à jour pour le

Detail-Type	Description	Unit	Source	Message
				cache <nom-cache>.
La limite de cache approche	ARN du cache	notification	serverless-cache	L'emplacement <X> utilise plus de <Y> % de la limite de 32 Go par emplacement. Par exemple, l'emplacement 10 utilise plus de 90 % de la limite de 32 Go par emplacement.
Échec de la mise à jour du cache	ARN du cache	échec	serverless-cache	La mise à jour des limites pour le cache <nom-cache> a échoué, car celui-ci a été supprimé.
Échec de la mise à jour du cache	ARN du cache	échec	serverless-cache	La mise à jour des limites pour le cache <nom-cache> a échoué en raison d'une configuration non valide.

Detail-Type	Description	Unit	Source	Message
Échec de la mise à jour du cache	ARN du cache	échec	serverless-cache	La mise à jour des limites pour le cache <nom-cache> a échoué, car les données actuellement mises en cache dépassent les nouvelles limites. Videz certaines données avant d'appliquer les limites.

Événements de capture instantanée du cache sans serveur (Valkey ou Redis OSS)

Detail-Type	Resources-list	Catégorie	Source	Message
Instantané créé	ARN du cache ARN de l'instantané	création	serverless-cache-snapshot	L'instantané <nom-instantané> a été créé pour le cache <nom-cache>.
Échec de la création de l'instantané	ARN du cache ARN de l'instantané	échec	serverless-cache-snapshot	Échec de la création de l'instantané du cache <nom-cache>. La création de l'instantané <nom-inst

Detail-Type	Resources-list	Catégorie	Source	Message
				antané> a échoué avec la clé gérée par le client <ID-clé>, <motif>. Messages relatifs au motif de l'échec : • car la clé gérée par le client est désactivée • car la clé gérée par le client est introuvable • car la demande a expiré

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de la création de l'instantané	ARN du cache ARN de l'instantané	échec	serverless-cache-snapshot	Échec de la création de l'instantané du cache <nom-cache>. La création de l'instantané <nom-instantané> a échoué <motif>. Motif par défaut : • en raison d'une erreur interne
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter le cliché vers le compartiment %s car il ElastiCache ne dispose pas des autorisations nécessaires pour accéder au compartiment.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car un objet portant le même nom existe déjà dans le compartiment.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car l'identifiant du compte propriétaire du compartiment a changé.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment S3 n'est pas accessible.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment n'est pas accessible.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment n'existe pas.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s » avec la clé gérée par le client %s de l'instantané source <motif>.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s ».
Échec de la copie de l'instantané	ARN-1 de l'instantané ARN-2 de l'instantané	échec	serverless-cache-snapshot	Échec de la copie de l'instantané <nom-instantané>. Impossible de copier l'instantané « %s » vers l'instantané « %s » avec la clé gérée par le client <ID-clé> de l'instantané source <nom-motif>.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de la copie de l'instantané	ARN-1 de l'instantané ARN-2 de l'instantané	échec	serverless-cache-snapshot	Échec de la copie de l'instantané <nom-instantané>. Impossible de copier l'instantané « %s » vers l'instantané « %s » avec la clé gérée par le client « %s » « %s » de l'instantané cible.

Mesures et événements de cluster conçus par vos soins pour Valkey et Redis OSS

ElastiCache propose une grande variété de métriques et d'événements pour surveiller les clusters conçus par vos soins lorsque vous travaillez avec Valkey et Redis OSS. Cela inclut les métriques au niveau de l'hôte, les métriques au niveau des commandes et les journaux d'événements disponibles via Amazon Simple Notification Service (SNS). AWS CLI

Rubriques

- Métriques pour les clusters auto-conçus
- Événements pour les clusters conçus par vos soins (Valkey et Redis OSS)

Métriques pour les clusters auto-conçus

Lorsque vous concevez vous-même des clusters, vous ElastiCache émettez des métriques au niveau de chaque nœud, notamment des métriques au niveau de l'hôte et des métriques du cache.

Pour plus d'informations sur les métriques au niveau de l'hôte, consultez Métriques au niveau de l'hôte.

Pour plus d'informations sur les métriques au niveau du nœud, consultez Métriques pour Valkey et Redis OSS.

Événements pour les clusters conçus par vos soins (Valkey et Redis OSS)

ElastiCache enregistre les événements liés aux caches que vous avez conçus vous-même. Lorsque vous travaillez avec des clusters conçus par vos soins, vous pouvez consulter les événements de votre cluster dans la ElastiCache console, à l'aide du AWS CLI ou à l'aide d'Amazon Simple Notification Service (SNS). Les événements de cluster conçus par vos soins ne sont pas publiés sur Amazon EventBridge.

Les informations des événements d'un cluster auto-conçu 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 l'événement. Vous pouvez facilement récupérer les événements du journal à l'aide de la ElastiCache console, de la commande AWS CLI describe-events ou de l'action ElastiCache API. DescribeEvents

Affichage ElastiCache des événements (console)

La procédure suivante affiche les événements à l'aide de la ElastiCache console.

Pour afficher les événements à l'aide de la ElastiCache console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/
2. Dans le volet de navigation, choisissez Événements pour consulter une liste de tous les événements disponibles.
3. Sur l'écran Événements, chaque ligne de la liste représente un événement et affiche sa source, son type, son heure GMT et sa description. 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.

Afficher ElastiCache les événements (AWS CLI)

Pour générer une liste d' ElastiCache événements 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 auto-conçu.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Le code suivant répertorie tous les événements des caches auto-conçus qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Événements pour les clusters auto-conçus

Cette section répertorie la liste des événements que vous pouvez vous attendre à recevoir pour vos clusters auto-conçus.

Les ElastiCache événements suivants déclenchent des notifications Amazon SNS. Pour de plus amples informations sur les détails de l'événement, veuillez consulter Affichage des ElastiCache événements.

Nom de l'événement	Message	Description
ElastiCache:AddCacheNodeComplete	`ElastiCache:AddCacheNodeComplete : cache-cluster`	Un nœud de cache a été ajouté au cluster de cache et est prêt à être utilisé.
ElastiCache: AddCacheNodeFailed en raison du nombre insuffisant d'adresses IP libres	`ElastiCache:AddCacheNodeFailed : cluster-name`	Un nœud de cache n'a pas pu être ajouté, car il n'y a pas suffisamment d'adresses IP disponibles.
ElastiCache:CacheClusterParametersChanged	`ElastiCache:CacheClusterParametersChanged : cluster-name`	Un ou plusieurs paramètres du cluster de cache ont été modifiés.
ElastiCache:CacheClusterProvisioningComplete	`ElastiCache:CacheClusterProvisioningComplete cluster-name-0001-005`	La mise en service d'un cluster de cache est terminée, et les nœuds de cache dans le cluster de cache sont prêts à être utilisés.
ElastiCache: CacheClusterProvisioningFailed en raison d'un état réseau incompatible	`ElastiCache:CacheClusterProvisioningFailed : cluster-name`	Il y a eu une tentative de lancement d'n nouveau cluster de cache dans un Virtual Private Cloud (VPC) inexistant.
ElastiCache:CacheClusterScalingComplete	`CacheClusterScalingComplete : cluster-name`	Dimensionnement du cluster de cache réalisé avec succès.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : *cluster-name*	Le dimensionnement du cluster de cache a échoué.
ElastiCache:CacheClusterSecurityGroupModified	`ElastiCache:CacheClusterSecurityGroupModified : cluster-name`	L'un des événements suivants s'est produit : •

Nom de l'événement	Message	Description
		La liste des groupes de sécurité de cache autorisés à accéder au cluster de cache a été modifiée. • Un ou plusieurs nouveaux groupes EC2 de sécurité ont été autorisés sur l'un des groupes de sécurité de cache associés au cluster de cache. • Un ou plusieurs groupes EC2 de sécurité ont été révoqués de l'un des groupes de sécurité de cache associés au cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceStarted	`ElastiCache:CacheNodeReplaceStarted : cluster-name`	ElastiCache a détecté que l'hôte exécutant un nœud de cache est dégradé ou inaccessible et a commencé à remplacer le nœud de cache. **Note** L'entrée DNS pour le nœud de cache remplacé n'est pas changée. 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 de cache peuvent cesser d'utiliser le nœud de cache même après ElastiCache son remplacement ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceComplete	`ElastiCache:CacheNodeReplaceComplete : cluster-name`	ElastiCache a détecté que l'hôte exécutant un nœud de cache est dégradé ou inaccessible et a terminé de remplacer le nœud de cache. **Note** L'entrée DNS pour le nœud de cache remplacé n'est pas changée. 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 de cache peuvent cesser d'utiliser le nœud de cache même après ElastiCache son remplacement ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.

Nom de l'événement	Message	Description
ElastiCache:CacheNodesRebooted	`ElastiCache:CacheNodesRebooted : cluster-name`	Un ou plusieurs nœuds de cache ont été redémarrés. Message (Memcached) : `"Cache node %s shutdown"` puis un second message : `"Cache node %s restarted"`
ElastiCache: CertificateRenewalComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:CertificateRenewalComplete`	Le certificat Amazon d'une autorité de certification a été renouvelé avec succès.
ElastiCache:CreateReplicationGroupComplete	`ElastiCache:CreateReplicationGroupComplete : cluster-name`	Le groupe de réplication a bien été créé.
ElastiCache:DeleteCacheClusterComplete	`ElastiCache:DeleteCacheClusterComplete : cluster-name`	La suppression d'un cluster de cache et de tous les nœuds de cache associés est terminée.
ElastiCache: FailoverComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:FailoverComplete : mycluster`	Basculement vers un nœud du réplica réussi.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	`ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : cluster-name-0001-005`	Le nombre de réplicas dans le cluster a été augmenté.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	`ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : cluster-name-0003-004`	Le processus d'ajout de réplicas à votre cluster a commencé.
ElastiCache:NodeReplacementCanceled	`ElastiCache:NodeReplacementCanceled : cluster-name`	Le remplacement d'un nœud de votre cluster qui était prévu a été annulé.
ElastiCache:NodeReplacementRescheduled	`ElastiCache:NodeReplacementRescheduled : cluster-name`	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 (Valkey et Redis OSS).
ElastiCache:NodeReplacementScheduled	`ElastiCache:NodeReplacementScheduled : cluster-name`	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 (Valkey et Redis OSS).
ElastiCache:RemoveCacheNodeComplete	`ElastiCache:RemoveCacheNodeComplete : cluster-name`	Un nœud de cache a été supprimé du cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupScalingComplete	`ElastiCache:ReplicationGroupScalingComplete :` *`cluster-name`*	L'opération d'extension du groupe de réplication a bien été effectuée.
ElastiCache:ReplicationGroupScalingFailed	`"Failed applying modification to cache node type to %s."`	L'opération d'extension du groupe de réplication a échoué.
ElastiCache:ServiceUpdateAvailableForNode	`"Service update is available for cache node %s."`	Une mise à jour en libre-service est disponible pour le nœud.
ElastiCache: SnapshotComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:SnapshotComplete :` *`cluster-name`*	Un instantané de cache s'est terminé avec succès.
ElastiCache: SnapshotFailed (Valkey ou Redis OSS uniquement)	`SnapshotFailed :` *`cluster-name`*	Un instantané de cache a échoué. Consultez les événements de cache du cluster pour en savoir plus. Si vous décrivez l'instantané, consultez `DescribeSnapshots`, l'état sera `failed`.

Métriques et événements pour Memcached

Cette section décrit les métriques et les événements que vous pouvez surveiller lorsque vous travaillez avec des caches sans serveur et conçus par Memcached.

Rubriques

- Métriques de cache sans serveur Memcached
- Événements de cache sans serveur Memcached

Métriques de cache sans serveur Memcached

Cette section décrit les métriques et les événements que vous pouvez surveiller lorsque vous travaillez avec des caches sans serveur Memcached.

L'espace de `AWS/ElastiCache` noms inclut les CloudWatch métriques suivantes pour vos caches sans serveur Memcached.

Métrique	Description	Unit
BytesUsedForCache	Nombre total d'octets utilisés par les données stockées dans votre cache.	Octets
ElastiCacheProcessingUnits	Le nombre total de ElastiCacheProcessingUnits (ECPUs) consommés par les requêtes exécutées sur votre cache	Nombre
SuccessfulReadRequestLatency	Latence des demandes de lecture réussies.	Microsecondes
SuccessfulWriteRequestLatency	Latence des demandes d'écriture réussies.	Microsecondes
TotalCmdsCount	Nombre total de commandes exécutées sur votre cache.	Nombre

Métrique	Description	Unit
CurrConnections	Nombre de connexions client à votre cache.	Nombre
ThrottledCmds	Le nombre de demandes qui ont été limitées ElastiCache parce que la charge de travail augmentait plus rapidement que ce que l' ElastiCache on pouvait faire.	Nombre
NewConnections	Nombre total de connexion s qui ont été acceptées par le serveur au cours de cette période.	Nombre
CurrItems	Nombre d'éléments dans le cache.	Nombre
NetworkBytesIn	Nombre total d'octets transféré s dans le cache.	Octets
NetworkBytesOut	Nombre total d'octets transféré s hors du cache.	Octets
Evictions	Nombre de clés expulsées par le cache.	Nombre
Reclaimed	Nombre de clés expirées par le cache.	Nombre

Métriques au niveau de la commande

ElastiCache émet également les métriques de niveau commande Memcached suivantes

Métrique	Description	Unit
CmdGet	Nombre de commandes get reçues par le cache.	Nombre
CmdSet	Nombre de commandes set reçues par le cache.	Nombre
CmdTouch	Nombre de commandes touch reçues par le cache.	Nombre
GetHits	Nombre de requêtes get reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
GetMisses	Nombre de requêtes get reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
IncrHits	Nombre de requêtes increment reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
IncrMisses	Nombre de requêtes increment reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
DecrHits	Nombre de requêtes decrement reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
DecrMisses	Nombre de requêtes decrement reçues par le	Nombre

Métrique	Description	Unit
	cache dans lesquelles la clé demandée était introuvable.	
DeleteHits	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
DeleteMisses	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
TouchHits	Nombre de clés qui ont été touchées et auxquelles une nouvelle date d'expiration a été affectée.	Nombre
TouchMisses	Nombre de clés qui ont été touchées, mais qui sont introuvables.	Nombre
CasHits	Nombre de demandes cas reçues par le cache dans lesquelles la clé demandée a été trouvée et la valeur cas correspondait.	Nombre
CasMisses	Nombre de demandes cas reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre

Métrique	Description	Unit
CasBadval	Nombre de demandes cas reçues par le cache dans lesquelles la valeur cas ne correspondait pas à la valeur cas stockée.	Nombre
CmdFlush	Nombre de commandes flush reçues par le cache.	Nombre

Événements de cache sans serveur Memcached

ElastiCache enregistre les événements liés à votre cache sans serveur. 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 ElastiCache console, de la commande AWS CLI describe-events ou de l'action ElastiCache API. `DescribeEvents`

Vous pouvez choisir de surveiller, d'ingérer, de transformer les ElastiCache événements et d'agir en conséquence à l'aide d'Amazon EventBridge. Pour en savoir plus, consultez le guide EventBridge de démarrage d'Amazon.

Affichage ElastiCache des événements (console)

Pour afficher les événements à l'aide de la ElastiCache console :

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à https://console.aws.amazon.com/elasticache/
2. Dans le volet de navigation, choisissez Événements pour consulter une liste de tous les événements disponibles.
3. Sur l'écran Événements, chaque ligne de la liste représente un événement et affiche sa source, son type, son heure GMT et sa description. 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.

Afficher ElastiCache les événements (AWS CLI)

Pour générer une liste d' ElastiCache événements 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 cache sans serveur.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Le code suivant répertorie tous les événements de caches sans serveur qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Événements pour les caches sans serveur

Cette section décrit les différents types d'événements que vous pouvez recevoir pour vos caches sans serveur.

Événements de création de caches sans serveur

Detail-Type	Description	Unit	Source	Message
Cache créé	ARN du cache	création	serverless-cache	Le cache <nom-cache> est créé et prêt à être utilisé.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Nombre d'adresses IP libres insuffisant pour créer un point de terminaison de VPC.

Detail-Type	Description	Unit	Source	Message
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Sous-réseaux non valides fournis dans la demande.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Limite de quota atteinte pour la création d'un point de terminaison de VPC.
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de la création du cache <nom-cache>. Vous n'êtes pas autorisé à créer un point de terminaison de VPC.

Événements de mise à jour du cache sans serveur (Memcached)

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	SecurityGroups mis à jour pour le cache<cache-name>.
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	Balises mises à jour pour le cache <nom-cache>.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-name>a échoué. SecurityGroups échec de la mise à jour.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-name>a échoué. SecurityGroups la mise à jour a échoué en raison d'autorisations insuffisantes.
Échec de la mise à jour du cache	ARN du cache	modification de configuration	serverless-cache	La mise à jour du cache <cache-name>a échoué. SecurityGroups la mise à jour a échoué car SecurityGroups

Detail-Type	Liste de ressources	Catégorie	Source	Message
				ils ne sont pas valides.

Événements de suppression du cache sans serveur (Memcached)

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache supprimé	ARN du cache	suppression	serverless-cache	Le cache <nom-cache> a été supprimé.

Événements de limite d'utilisation du cache sans serveur (Memcached)

Detail-Type	Description	Unit	Source	Message
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	Limites mises à jour pour le cache <nom-cache>.
Échec de la mise à jour du cache	ARN du cache	échec	serverless-cache	La mise à jour des limites pour le cache <nom-cache> a échoué, car celui-ci a été supprimé.
Échec de la mise à jour du cache	ARN du cache	échec	serverless-cache	La mise à jour des limites pour le cache <nom-cache> a échoué

Detail-Type	Description	Unit	Source	Message
				en raison d'une configuration non valide.

Événements de capture instantanée du cache sans serveur (Memcached)

Detail-Type	Resources-list	Catégorie	Source	Message
Instantané créé	ARN du cache ARN de l'instantané	création	serverless-cache-snapshot	L'instantané <nom-instantané> a été créé pour le cache <nom-cache>.
Échec de la création de l'instantané	ARN du cache ARN de l'instantané	échec	serverless-cache-snapshot	Échec de la création de l'instantané du cache <nom-cache>. La création de l'instantané <nom-instantané> a échoué avec la clé gérée par le client <ID-clé>, <motif>. Messages relatifs au motif de l'échec : • car la clé gérée par

Detail-Type	Resources-list	Catégorie	Source	Message
				le client est désactivée • car la clé gérée par le client est introuvable • car la demande a expiré
Échec de la création de l'instantané	ARN du cache ARN de l'instantané	échec	serverless-cache-snapshot	Échec de la création de l'instantané du cache <nom-cache>. La création de l'instantané <nom-instantané> a échoué <motif>. Motif par défaut : • en raison d'une erreur interne

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter le cliché vers le compartiment %s car il ElastiCache ne dispose pas des autorisations nécessaires pour accéder au compartiment.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car un objet portant le même nom existe déjà dans le compartiment.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car l'identifiant du compte propriétaire du compartiment a changé.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment S3 n'est pas accessible.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment n'est pas accessible.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s », car le compartiment n'existe pas.

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s » avec la clé gérée par le client %s de l'instantané source <motif>.
Échec de l'exportation de l'instantané	ARN de l'instantané	échec	serverless-cache-snapshot	Échec de l'exportation de l'instantané du cache <nom-cache>. Impossible d'exporter l'instantané vers le compartiment « %s ».

Detail-Type	Resources-list	Catégorie	Source	Message
Échec de la copie de l'instantané	ARN-1 de l'instantané ARN-2 de l'instantané	échec	serverless-cache-snapshot	Échec de la copie de l'instantané <nom-instantané>. Impossible de copier l'instantané « %s » vers l'instantané « %s » avec la clé gérée par le client <ID-clé> de l'instantané source <nom-motif>.
Échec de la copie de l'instantané	ARN-1 de l'instantané ARN-2 de l'instantané	échec	serverless-cache-snapshot	Échec de la copie de l'instantané <nom-instantané>. Impossible de copier l'instantané « %s » vers l'instantané « %s » avec la clé gérée par le client « %s » « %s » de l'instantané cible.

Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail

Amazon ElastiCache est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service sur Amazon ElastiCache. CloudTrail capture tous les appels d'API pour Amazon ElastiCache sous forme d'événements, y compris les appels depuis la ElastiCache console Amazon et les appels de code vers les opérations ElastiCache d'API Amazon. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon ElastiCache. 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 Amazon ElastiCache, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

ElastiCache Informations Amazon dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité a lieu sur Amazon ElastiCache, 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 à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour Amazon ElastiCache, créez un parcours. 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 ElastiCache actions Amazon sont enregistrées CloudTrail et documentées dans la référence ElastiCache d'API. Par exemple, les appels au`CreateCacheCluster`, `DescribeCacheCluster` et les `ModifyCacheCluster` 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 la section Élément userIdentity CloudTrail .

Comprendre les entrées des fichiers ElastiCache journaux Amazon

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'`CreateCacheCluster`action.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
```

```
    },
    "eventTime":"2014-12-01T22:00:35Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"CreateCacheCluster",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "numCacheNodes":2,
        "cacheClusterId":"test-memcached",
        "engine":"memcached",
        "aZMode":"cross-az",
        "cacheNodeType":"cache.m1.small",

    },
    "responseElements":{
        "engine":"memcached",
        "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
        "cacheParameterGroup":{
            "cacheParameterGroupName":"default.memcached1.4",
            "cacheNodeIdsToReboot":{
            },
            "parameterApplyStatus":"in-sync"
        },
        "preferredAvailabilityZone":"Multiple",
        "numCacheNodes":2,
        "cacheNodeType":"cache.m1.small",

        "cacheClusterStatus":"creating",
        "autoMinorVersionUpgrade":true,
        "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
        "cacheClusterId":"test-memcached",
        "engineVersion":"1.4.14",
        "cacheSecurityGroups":[
            {
                "status":"active",
                "cacheSecurityGroupName":"default"
            }
        ],
        "pendingModifiedValues":{
        }
    },
    "requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
```

```
    "eventID":"92762127-7a68-42ce-8787-927d2174cde1"
}
```

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`DescribeCacheCluster`action. Notez que pour tous les appels Amazon ElastiCache Describe (`Describe*`), la `ResponseElements` section est supprimée et apparaît sous la forme`null`.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
    },
    "eventTime":"2014-12-01T22:01:00Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"DescribeCacheClusters",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "showCacheNodeInfo":false,
        "maxRecords":100
    },
    "responseElements":null,
    "requestID":"1f0b5031-3548-11e4-9376-c1d979ba565a",
    "eventID":"a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

L'exemple suivant montre une entrée de CloudTrail journal qui enregistre une `ModifyCacheCluster` action.

```
{
    "eventVersion":"1.01",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"EXAMPLEEXAMPLEEXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
        "accountId":"123456789012",
```

```
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"elasticache-allow"
    },
    "eventTime":"2014-12-01T22:32:21Z",
    "eventSource":"elasticache.amazonaws.com",
    "eventName":"ModifyCacheCluster",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
    "requestParameters":{
        "applyImmediately":true,
        "numCacheNodes":3,
        "cacheClusterId":"test-memcached"
    },
    "responseElements":{
        "engine":"memcached",
        "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
        "cacheParameterGroup":{
            "cacheParameterGroupName":"default.memcached1.4",
            "cacheNodeIdsToReboot":{
            },
            "parameterApplyStatus":"in-sync"
        },
        "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
        "preferredAvailabilityZone":"Multiple",
        "numCacheNodes":2,
        "cacheNodeType":"cache.m1.small",
        "cacheClusterStatus":"modifying",
        "autoMinorVersionUpgrade":true,
        "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
        "cacheClusterId":"test-memcached",
        "engineVersion":"1.4.14",
        "cacheSecurityGroups":[
            {
                "status":"active",
                "cacheSecurityGroupName":"default"
            }
        ],
        "configurationEndpoint":{
            "address":"test-memcached.example.cfg.use1prod.cache.amazonaws.com",
            "port":11211
        },
        "pendingModifiedValues":{
```

```
            "numCacheNodes":3
        }
    },
    "requestID":"807f4bc3-354c-11e4-9376-c1d979ba565a",
    "eventID":"e9163565-376f-4223-96e9-9f50528da645"
}
```

Surveillance des événements par Amazon SNS ElastiCache

Lorsque des événements importants se produisent pour un cluster, ElastiCache envoie une notification à une rubrique 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 ElastiCache notifications Amazon SNS
- Affichage des ElastiCache événements
- Notifications d'événements Amazon SNS

Gestion des ElastiCache notifications Amazon SNS

Vous pouvez configurer ElastiCache l'envoi de notifications pour les événements importants du cluster à l'aide d'Amazon Simple Notification Service (Amazon SNS). Dans ces exemples, vous allez configurer un cluster avec l'Amazon Resource Name (ARN) d'une rubrique Amazon SNS pour recevoir des notifications.

> **Note**
>
> - Cette rubrique suppose que vous êtes inscrit à Amazon SNS, que vous avez souscrit à une rubrique Amazon SNS et que vous l'avez configurée. Pour plus d'informations sur Amazon SNS, veuillez consulter le Guide du développeur d'Amazon Simple Notification Service.
> - Par défaut, `API modify-replication-group` cela affecte tous les groupes d'une région et pas seulement le groupe actuellement spécifié. Si vous souhaitez configurer un groupe spécifique dans une région différemment des autres groupes, vous pouvez utiliser

l'`--notification-topic-arn`option permettant de créer une rubrique distincte pour ce groupe.

Ajout d'une rubrique Amazon SNS

Les sections suivantes expliquent comment ajouter une rubrique Amazon SNS à l'aide de la AWS console, de l'API ou de l' AWS CLI API. ElastiCache

Ajout d'une rubrique Amazon SNS (console)

La procédure suivante vous indique comment ajouter une rubrique Amazon SNS pour un cluster. Lorsque vous utilisez Valkey ou Redis OSS pour ajouter une rubrique Amazon SNS pour un groupe de réplication à l'étape 2, au lieu de choisir un cluster, choisissez un groupe de réplication. Suivez ensuite les mêmes étapes restantes.

> **Note**
>
> Ce processus permet également de modifier la rubrique Amazon SNS.

Pour ajouter ou modifier une rubrique Amazon SNS pour un cluster (console)

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans Clusters, choisissez le cluster pour lequel vous souhaitez ajouter ou modifier un ARN de rubrique Amazon SNS.
3. Sélectionnez Modifier.
4. Dans Modify Cluster (Modifier le cluster) sous Topic for SNS Notification (Rubrique pour notification SNS), choisissez la rubrique SNS que vous voulez ajouter ou choisissez Manual ARN input (Saisie d'ARN manuelle) et tapez l'ARN de la rubrique Amazon SNS.
5. Sélectionnez Modifier.

Ajout d'une rubrique Amazon SNS AWS CLI)

Pour ajouter ou modifier une rubrique Amazon SNS pour un cluster, utilisez la AWS CLI commande. `modify-cache-cluster`

L'exemple de code suivant ajoute un ARN de rubrique Amazon SNS à my-cluster.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --notification-topic-arn arn:aws:sns:us-
west-2:123456789xxx:ElastiCacheNotifications
```

Pour Windows :

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Pour de plus amples informations, veuillez consulter modify-cache-cluster.

Ajouter une rubrique Amazon SNS (API) ElastiCache

Pour ajouter ou modifier une rubrique Amazon SNS pour un cluster, appelez l'action `ModifyCacheCluster` avec les paramètres suivants :

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=false
    &CacheClusterId=my-cluster
    &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-
west-2%3A565419523791%3AElastiCacheNotifications
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
```

```
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Pour de plus amples informations, veuillez consulter ModifyCacheCluster.

Activation et désactivation des notifications Amazon SNS

Vous pouvez activer ou désactiver les notifications pour un cluster. Les procédures suivantes vous expliquent comment désactiver les notifications Amazon SNS.

Activation et désactivation des notifications Amazon SNS (console)

Pour désactiver les notifications Amazon SNS à l'aide du AWS Management Console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Pour consulter une liste de vos clusters exécutant Memcached, dans le volet de navigation, sélectionnez Memcached.

   Pour voir la liste de vos clusters exécutant Valkey ou Redis OSS, dans le volet de navigation, choisissez Valkey ou Redis OSS.
3. Cochez la case à gauche du cluster pour lequel vous souhaitez modifier la notification.
4. Sélectionnez Modifier.
5. Dans Modifier le cluster sous Rubrique pour notification SNS, choisissez Désactiver les notifications.
6. Sélectionnez Modifier.

Activation et désactivation des notifications Amazon SNS (AWS CLI)

Pour désactiver les notifications Amazon SNS, utilisez la commande `modify-cache-cluster` avec les paramètres suivants :

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id my-cluster \
    --notification-topic-status inactive
```

Pour Windows :

```
aws elasticache modify-cache-cluster ^
    --cache-cluster-id my-cluster ^
    --notification-topic-status inactive
```

> **Note**
>
> Lorsque le cluster de cache appartient à un groupe de réplication, vous devez utiliser la commande CLI `modify-replication-group` pour activer ou désactiver les notifications SNS.

Activation et désactivation des notifications ElastiCache Amazon SNS (API)

Pour désactiver les notifications Amazon SNS, appelez l'action `ModifyCacheCluster` avec les paramètres suivants :

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Cet appel vous renvoie des informations semblables à ce qui suit :

Example

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=ModifyCacheCluster
    &ApplyImmediately=false
    &CacheClusterId=my-cluster
    &NotificationTopicStatus=inactive
    &Version=2014-12-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Date=20141201T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Affichage des ElastiCache événements

ElastiCache enregistre les événements liés à vos instances de cluster, à vos groupes de sécurité et à vos 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 ElastiCache console, de la AWS CLI `describe-events` commande ou de l'action de l' ElastiCache API`DescribeEvents`.

Les procédures suivantes vous montrent comment afficher tous les ElastiCache événements des dernières 24 heures (1 440 minutes).

Affichage ElastiCache des événements (console)

La procédure suivante affiche les événements à l'aide de la ElastiCache console.

Pour afficher les événements à l'aide de la ElastiCache console

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Dans le volet de navigation, choisissez Événements pour consulter une liste de tous les événements disponibles.

   Sur l'écran Événements, chaque ligne de la liste représente un événement et affiche la source de l'événement, le type d'événement (cache-cluster cache-parameter-group, cache-security-group, ou cache-subnet-group), l'heure GMT de l'événement et une 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.

Afficher ElastiCache les événements (AWS CLI)

Pour générer une liste d' ElastiCache événements 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 de cache.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Le code suivant répertorie tous les événements qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

La sortie de la commande `describe-events` ressemble à ceci.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
{
    "Events": [
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Finished modifying number of nodes from 1 to 3",
            "Date": "2020-06-09T02:01:21.772Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0002 in availability zone us-west-2a",
            "Date": "2020-06-09T02:01:21.716Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0003 in availability zone us-west-2a",
            "Date": "2020-06-09T02:01:21.706Z"
        },
        {
            "SourceIdentifier": "my-mem-cluster",
            "SourceType": "cache-cluster",
            "Message": "Increasing number of requested nodes",
            "Date": "2020-06-09T01:58:34.178Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2c",
            "Date": "2020-06-09T01:51:14.120Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
```

```
            "Message": "This cache cluster does not support persistence (ex:
 'appendonly').  Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:51:14.095Z"
        },
        {
            "SourceIdentifier": "mycluster-0003-004",
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:51:14.094Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2b",
            "Date": "2020-06-09T01:42:55.603Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "This cache cluster does not support persistence (ex:
 'appendonly').  Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:42:55.576Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-005",
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:42:55.574Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
            "SourceType": "cache-cluster",
            "Message": "Added cache node 0001 in availability zone us-west-2b",
            "Date": "2020-06-09T01:28:40.798Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
            "SourceType": "cache-cluster",
            "Message": "This cache cluster does not support persistence (ex:
 'appendonly').  Please use a different instance type to enable persistence.",
            "Date": "2020-06-09T01:28:40.775Z"
        },
        {
            "SourceIdentifier": "mycluster-0001-004",
```

```
            "SourceType": "cache-cluster",
            "Message": "Cache cluster created",
            "Date": "2020-06-09T01:28:40.773Z"
        }
    ]
}
```

Pour plus d'informations, notamment sur les paramètres disponibles et les valeurs de paramètre autorisées, consultez `describe-events`.

Affichage ElastiCache des événements (ElastiCache API)

Pour générer une liste d' ElastiCache événements à l'aide de l' ElastiCache API, utilisez l'`DescribeEvents`action. 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 derniers événements cache-cluster.

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeEvents
   &MaxRecords=40
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cache-cluster
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Le code suivant répertorie les événements de clusters de cache qui ont eu lieu au cours des dernières 24 heures (1440 minutes).

```
https://elasticache.us-west-2.amazonaws.com/
   ?Action=DescribeEvents
   &Duration=1440
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cache-cluster
   &Timestamp=20150202T192317Z
   &Version=2015-02-02
   &X-Amz-Credential=<credential>
```

Les actions ci-dessus doivent produire un résultat similaire à ce qui suit :

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">
    <DescribeEventsResult>
        <Events>
            <Event>
                <Message>Cache cluster created</Message>
                <SourceType>cache-cluster</SourceType>
                <Date>2015-02-02T18:22:18.202Z</Date>
                <SourceIdentifier>mem01</SourceIdentifier>
            </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

ElastiCache peut publier des messages à l'aide d'Amazon Simple Notification Service (SNS) lorsque des événements importants se produisent sur un cluster de cache. Cette fonctionnalité permet d'actualiser les listes de serveurs sur les ordinateurs clients connectés aux points de terminaison des nœuds de cache individuels de chaque cluster de cache.

> **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 ElastiCache notifications.

- Le AWS compte propriétaire de la rubrique Amazon SNS doit être le même que celui qui possède le cluster de cache sur lequel les notifications sont activées.
- La rubrique Amazon SNS dans laquelle vous publiez ne peut pas être chiffrée.

> **Note**
>
> Il est possible d'attacher une rubrique Amazon SNS chiffrée (au repos) au cluster. Cependant, le statut du sujet affiché sur la ElastiCache console sera indiqué comme inactif, ce qui dissocie effectivement le sujet du cluster lorsque des messages sont ElastiCache envoyés vers le sujet.

- Le sujet Amazon SNS doit se trouver dans la même région que le ElastiCache cluster.

ElastiCache Événements

Les ElastiCache événements suivants déclenchent des notifications Amazon SNS. Pour de plus amples informations sur les détails de l'événement, veuillez consulter Affichage des ElastiCache événements.

Nom de l'événement	Message	Description
ElastiCache:AddCacheNodeComplete	`ElastiCache:AddCacheNodeComplete : cache-cluster`	Un nœud de cache a été ajouté au cluster de cache et est prêt à être utilisé.
ElastiCache: AddCacheNodeFailed en raison d'un nombre insuffisant d'adresses IP libres	`ElastiCache:AddCacheNodeFailed : cluster-name`	Un nœud de cache n'a pas pu être ajouté, car il n'y a pas suffisamment d'adresses IP disponibles.
ElastiCache:CacheClusterParametersChanged	`ElastiCache:CacheClusterParametersChanged : cluster-name`	Un ou plusieurs paramètres du cluster de cache ont été modifiés.
ElastiCache:CacheClusterProvisioningComplete	`ElastiCache:CacheClusterProvisioningComplete cluster-name-0001-005`	La mise en service d'un cluster de cache est terminée, et les nœuds de cache dans le

Nom de l'événement	Message	Description
		cluster de cache sont prêts à être utilisés.
ElastiCache: CacheClusterProvisioningFailed en raison d'un état réseau incompatible	`ElastiCache:CacheClusterProvisioningFailed : cluster-name`	Il y a eu une tentative de lancement d'n nouveau cluster de cache dans un Virtual Private Cloud (VPC) inexistant.
ElastiCache:CacheClusterScalingComplete	`CacheClusterScalingComplete : cluster-name`	Dimensionnement du cluster de cache réalisé avec succès.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : *cluster-name*	Le dimensionnement du cluster de cache a échoué.

Nom de l'événement	Message	Description
ElastiCache:CacheClusterSecurityGroupModified	`ElastiCache:CacheClusterSecurityGroupModified : cluster-name`	L'un des événements suivants s'est produit : • La liste des groupes de sécurité de cache autorisés à accéder au cluster de cache a été modifiée. • Un ou plusieurs nouveaux groupes EC2 de sécurité ont été autorisés sur l'un des groupes de sécurité de cache associés au cluster de cache. • Un ou plusieurs groupes EC2 de sécurité ont été révoqués de l'un des groupes de sécurité de cache associés au cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceStarted	`ElastiCache:CacheNodeReplaceStarted : cluster-name`	ElastiCache a détecté que l'hôte exécutant un nœud de cache est dégradé ou inaccessible et a commencé à remplacer le nœud de cache. **Note** L'entrée DNS pour le nœud de cache remplacé n'est pas changée. 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 de cache peuvent cesser d'utiliser le nœud de cache même après ElastiCache son remplacement ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceComplete	`ElastiCache:CacheNodeReplaceComplete : cluster-name`	ElastiCache a détecté que l'hôte exécutant un nœud de cache est dégradé ou inaccessible et a terminé de remplacer le nœud de cache. **Note** L'entrée DNS pour le nœud de cache remplacé n'est pas changée. 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 de cache peuvent cesser d'utiliser le nœud de cache même après ElastiCache son remplacement ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.

Nom de l'événement	Message	Description
ElastiCache:CacheNodesRebooted	`ElastiCache:CacheNodesRebooted : cluster-name`	Un ou plusieurs nœuds de cache ont été redémarrés. Message (Memcached) : `"Cache node %s shutdown"` puis un second message : `"Cache node %s restarted"`
ElastiCache: CertificateRenewalComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:CertificateRenewalComplete`	Le certificat Amazon d'une autorité de certification a été renouvelé avec succès.
ElastiCache:CreateReplicationGroupComplete	`ElastiCache:CreateReplicationGroupComplete : cluster-name`	Le groupe de réplication a bien été créé.
ElastiCache:DeleteCacheClusterComplete	`ElastiCache:DeleteCacheClusterComplete : cluster-name`	La suppression d'un cluster de cache et de tous les nœuds de cache associés est terminée.
ElastiCache: FailoverComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:FailoverComplete : mycluster`	Basculement vers un nœud du réplica réussi.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	`ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : cluster-name-0001-005`	Le nombre de réplicas dans le cluster a été augmenté.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	`ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : cluster-name-0003-004`	Le processus d'ajout de réplicas à votre cluster a commencé.
ElastiCache:NodeReplacementCanceled	`ElastiCache:NodeReplacementCanceled : cluster-name`	Le remplacement d'un nœud de votre cluster qui était prévu a été annulé.
ElastiCache:NodeReplacementRescheduled	`ElastiCache:NodeReplacementRescheduled : cluster-name`	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 (Valkey et Redis OSS).
ElastiCache:NodeReplacementScheduled	`ElastiCache:NodeReplacementScheduled : cluster-name`	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 (Valkey et Redis OSS).
ElastiCache:RemoveCacheNodeComplete	`ElastiCache:RemoveCacheNodeComplete : cluster-name`	Un nœud de cache a été supprimé du cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupScalingComplete	`ElastiCache:ReplicationGroupScalingComplete : cluster-name`	L'opération d'extension du groupe de réplication a bien été effectuée.
ElastiCache:ReplicationGroupScalingFailed	`"Failed applying modification to cache node type to %s."`	L'opération d'extension du groupe de réplication a échoué.
ElastiCache:ServiceUpdateAvailableForNode	`"Service update is available for cache node %s."`	Une mise à jour en libre-service est disponible pour le nœud.
ElastiCache: SnapshotComplete (Valkey ou Redis OSS uniquement)	`ElastiCache:SnapshotComplete : cluster-name`	Un instantané de cache s'est terminé avec succès.
ElastiCache: SnapshotFailed (Valkey ou Redis OSS uniquement)	`SnapshotFailed : cluster-name`	Un instantané de cache a échoué. Consultez les événements de cache du cluster pour en savoir plus. Si vous décrivez l'instantané, consultez `DescribeSnapshots`, l'état sera `failed`.

Rubriques en relation

- Affichage des ElastiCache événements

Livraison de journaux

> **Note**
>
> Slow Log est pris en charge pour Valkey 7.x et versions ultérieures, ainsi que pour les clusters de cache et les groupes de réplication Redis OSS utilisant la version 6.0 du moteur et les versions ultérieures.
> Le journal du moteur est pris en charge pour Valkey 7.x et versions ultérieures, ainsi que pour les clusters de cache et les groupes de réplication Redis OSS utilisant la version 6.2 du moteur et les versions ultérieures.

La diffusion du journal vous permet de diffuser SLOWLOG ou Engine Log vers l'une des deux destinations suivantes :

- Amazon Data Firehose
- Amazon CloudWatch Logs

Vous activez et configurez la livraison des journaux lorsque vous créez ou modifiez un cluster à l'aide de ElastiCache APIs. Chaque entrée de journal sera livrée à la destination spécifiée dans l'un des deux formats suivants :JSON ou TEXT.

Un nombre fixe d'entrées de journal lent sont extraites périodiquement du moteur. En fonction de la valeur spécifiée pour le paramètre moteur `slowlog-max-len`, des entrées de journal lentes supplémentaires peuvent ne pas être envoyées à la destination.

Vous pouvez choisir de modifier les configurations de diffusion ou de désactiver la livraison des journaux à tout moment à l'aide de la AWS console ou de l'une des APIs options de modification modify-cache-clusterou modify-replication-group.

Vous devez définir le paramètre `apply-immediately` pour toutes les modifications de diffusion des journaux.

> **Note**
>
> CloudWatch Les frais Amazon Logs s'appliquent lorsque la livraison des journaux est activée, même lorsque les journaux sont envoyés directement à Amazon Data Firehose. Pour plus d'informations, consultez la section Vended Logs dans Amazon CloudWatch Pricing.

Contenu d'une entrée de journal lente

Le journal lent contient les informations suivantes :

- CacheClusterId— L'ID du cluster de cache
- CacheNodeId— L'ID du nœud de cache
- Id – Identificateur progressif unique pour chaque entrée de journal lente
- Horodatage – Horodatage Unix à partir duquel la commande journalisée a été traitée
- Durée – Le temps nécessaire à son exécution, en microsecondes
- Commande – Commande utilisée par le client. Par exemple, `set foo bar` où se `foo` trouve la clé et où `bar` se trouve la valeur. ElastiCache remplace le nom et la valeur réels de la clé par `(2 more arguments)` pour éviter d'exposer des données sensibles.
- ClientAddress— Adresse IP et port du client
- ClientName— Nom du client s'il est défini via la `CLIENT SETNAME` commande

Contenu d'une entrée dans le journal du moteur

Le journal ElastiCache du moteur contient les informations suivantes :

- CacheClusterId— L'ID du cluster de cache
- CacheNodeId— L'ID du nœud de cache
- Niveau de journalisation : LogLevel peut être l'un des suivants :`VERBOSE("-")`,`NOTICE("*")`,`WARNING("#")`.
- Heure – L'heure UTC du message journalisé. L'heure est au format suivant : `"DD MMM YYYY hh:mm:ss.ms UTC"`
- Rôle – Rôle du nœud à partir duquel le journal est émis. Il peut s'agir de l'une des options suivantes : « M » pour Primary, « S » pour la réplique, « C » pour Writer Child Process on working on Writer Child RDB/AOF ou « X » pour Sentinel.
- Message : message du journal du moteur.

Autorisations pour configurer la journalisation

Vous devez inclure les autorisations IAM suivantes dans votre politique IAM user/role :

- `logs:CreateLogDelivery`

- `logs:UpdateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`

Pour plus d'informations, veuillez consulter Présentation de la gestion des accès : autorisations et politiques.

Spécifications du type de journal et du format de journal

Journal lent

Le journal lent prend en charge JSON et TEXT

L'exemple suivant illustre un rapport au format JSON :

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

L'exemple suivant illustre un rapport au format TEXT :

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
 arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Journal du moteur

Le journal du moteur prend en charge à la fois JSON et TEXT

L'exemple suivant illustre un rapport au format JSON :

```
{
```

```
  "CacheClusterId": "xxxxxxxxxzy-engine-log-test",
  "CacheNodeId": "0001",
  "LogLevel": "VERBOSE",
  "Role": "M",
  "Time": "12 Nov 2020 01:28:57.994 UTC",
  "Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.
 Check back later"
}
```

L'exemple suivant illustre un rapport au format TEXT :

```
xxxxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua
 script detected that is still in execution after 10000 milliseconds.
```

ElastiCache destinations de journalisation

Cette section décrit les destinations de journalisation que vous pouvez choisir pour vos ElastiCache journaux. Chaque section fournit des conseils pour la configuration de la journalisation pour le type de destination et des informations sur tout comportement spécifique au type de destination. Après avoir configuré votre destination de journalisation, vous pouvez fournir ses spécifications à la configuration de ElastiCache journalisation pour commencer à vous y connecter.

Rubriques

- Amazon CloudWatch Logs
- Amazon Data Firehose

Amazon CloudWatch Logs

- Vous spécifiez un groupe de CloudWatch journaux dans lequel les journaux seront livrés.
- Les journaux provenant de plusieurs clusters et groupes de réplication Valkey ou Redis OSS peuvent être fournis au même groupe de journaux.
- Un nouveau flux de logs sera créé pour chaque nœud au sein d'un cluster de cache ou d'un groupe de réplication et les logs seront diffusés aux flux de logs respectifs. Le nom du flux de journal utilisera le format suivant : `elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

Autorisations pour publier des journaux dans CloudWatch Logs

Vous devez disposer des paramètres d'autorisation suivants pour configurer l'envoi ElastiCache de journaux à un groupe de CloudWatch journaux de journaux :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "ElastiCacheLogging"
        },
        {
            "Sid": "ElastiCacheLoggingCWL",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Pour plus d'informations, consultez la section Logs envoyés à CloudWatch Logs.

Amazon Data Firehose

- Vous spécifiez un flux de diffusion Firehose dans lequel les logs seront livrés.
- Les journaux provenant de plusieurs clusters et groupes de réplication Valkey ou Redis OSS peuvent être transmis au même flux de diffusion.
- Les journaux de chaque nœud d'un cluster de cache ou d'un groupe de réplication seront diffusés au même flux de diffusion. Vous pouvez distinguer les messages de journal des différents nœuds de cache en fonction de la `cache-cluster-id` et `cache-node-id` inclus dans chaque message de journal.
- La livraison de journaux à Firehose n'est actuellement pas disponible dans la région Asie-Pacifique (Osaka).

Autorisations pour publier des journaux sur Firehose

Vous devez disposer des autorisations suivantes pour configurer l'envoi ElastiCache de journaux vers un flux de diffusion Amazon Kinesis Data Firehose.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "ElastiCacheLogging"
        },
        {
            "Sid": "ElastiCacheLoggingFHSLR",
            "Action": [
```

```
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "ElastiCacheLoggingFH",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "Amazon Kinesis Data Firehose delivery stream ARN",
            "Effect": "Allow"
        }
    ]
}
```

Spécification de la diffusion des journaux à l'aide de la console

À l'aide du, AWS Management Console vous pouvez créer un cluster Valkey ou Redis OSS (mode cluster désactivé) en suivant les étapes décrites dans Création d'un cluster Valkey (mode cluster désactivé) (console) ou créer un cluster Valkey ou Redis OSS (mode cluster activé) en suivant les étapes décrites dans. Création d'un cluster Valkey ou Redis OSS (mode cluster activé) (console) Dans les deux cas, vous configurez la diffusion des journaux en procédant comme suit :

1. Sous Paramètres avancés, choisissez Logs, puis cochez Slow logs ou Engine logs.
2. Sous Log format (Format de journal), sélectionnez Text (Texte) ou JSON.
3. Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
4. Sous Destination du journal, choisissez Create new et entrez le nom de votre bucket Amazon S3, le nom de votre groupe de CloudWatchLogs journaux ou le nom de votre flux Kinesis Data Firehose, ou choisissez Select existing, puis choisissez CloudWatch le nom de votre groupe de logs ou le nom de votre flux Kinesis Data Firehose,

Quand modifier un cluster :

Vous pouvez choisir d' enable/disable enregistrer la livraison ou de modifier le type, le format ou la destination de destination :

1. Connectez-vous à la console et ouvrez-la à l' ElastiCache adresse https://console.aws.amazon.com/elasticache/.

2. Dans le volet de navigation, choisissez Valkey clusters ou Redis OSS clusters.
3. Dans la liste des clusters, choisissez le cluster que vous souhaitez augmenter. Cliquez sur l'onglet Cluster name (Nom du cluster) et pas la case à cocher à côté.
4. Sur la page Cluster name (Nom du cluster), choisissez l'onglet Logs (Journaux).
5. Pour enable/disable ralentir les journaux, choisissez Activer les journaux lents ou Désactiver les journaux lents.
6. Pour les journaux enable/disable du moteur, choisissez Activer les journaux du moteur ou Désactiver les journaux du moteur.
7. Pour modifier votre configuration, choisissez Modify slow logs (Modifier les journaux lents) ou Modify engine logs (Modifier les journaux du moteur) :
   - Sous Type de destination, sélectionnez CloudWatch Logs ou Kinesis Firehose.
   - Sous Destination du journal, choisissez Create new et entrez le nom de votre groupe de CloudWatchLogs journaux ou le nom de votre flux Kinesis Data Firehose. Vous pouvez également choisir Select existing, puis le nom de votre groupe de CloudWatchLogs journaux ou le nom de votre flux Kinesis Data Firehose.

Spécifier la livraison du journal à l'aide du AWS CLI

Journal lent

Créez un groupe de réplication dont la transmission des CloudWatch journaux à Logs est lente.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
    --replication-group-id test-slow-log \
    --replication-group-description test-slow-log \
    --engine redis \
    --cache-node-type cache.r5.large \
    --num-cache-clusters 2 \
    --log-delivery-configurations '{
        "LogType":"slow-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
```

```
        "LogFormat":"json"
      }'
```

Pour Windows :

```
aws elasticache create-replication-group ^
    --replication-group-id test-slow-log ^
    --replication-group-description test-slow-log ^
    --engine redis ^
    --cache-node-type cache.r5.large ^
    --num-cache-clusters 2 ^
    --log-delivery-configurations '{
        "LogType":"slow-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Modifier un groupe de réplication pour transmettre un journal lent à CloudWatch Logs

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "DestinationType":"cloudwatch-logs",
      "DestinationDetails":{
        "CloudWatchLogsDetails":{

          "LogGroup":"my-log-group"
        }
      },
      "LogFormat":"json"
    }'
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "DestinationType":"cloudwatch-logs",
      "DestinationDetails":{
        "CloudWatchLogsDetails":{
          "LogGroup":"my-log-group"
        }
      },
      "LogFormat":"json"
    }'
```

Modifier un groupe de réplication pour désactiver la diffusion de journaux slow

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "Enabled":false
    }'
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"slow-log",
      "Enabled":false
    }'
```

Journal du moteur

Créez un groupe de réplication avec livraison des journaux du moteur à CloudWatch Logs.

Pour Linux, macOS ou Unix :

```
aws elasticache create-replication-group \
    --replication-group-id test-slow-log \
    --replication-group-description test-slow-log \
    --engine redis \
    --cache-node-type cache.r5.large \
    --num-cache-clusters 2 \
    --log-delivery-configurations '{
        "LogType":"engine-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Pour Windows :

```
aws elasticache create-replication-group ^
    --replication-group-id test-slow-log ^
    --replication-group-description test-slow-log ^
    --engine redis ^
    --cache-node-type cache.r5.large ^
    --num-cache-clusters 2 ^
    --log-delivery-configurations '{
        "LogType":"engine-log",
        "DestinationType":"cloudwatch-logs",
        "DestinationDetails":{
          "CloudWatchLogsDetails":{
            "LogGroup":"my-log-group"
          }
        },
        "LogFormat":"json"
      }'
```

Modifier un groupe de réplication pour transmettre le journal du moteur à Firehose

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
```

```
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "DestinationType":"kinesis-firehose",
      "DestinationDetails":{
      "KinesisFirehoseDetails":{
         "DeliveryStream":"test"
       }
     },
      "LogFormat":"json"
    }'
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "DestinationType":"kinesis-firehose",
      "DestinationDetails":{
      "KinesisFirehoseDetails":{
         "DeliveryStream":"test"
       }
      },
      "LogFormat":"json"
    }'
```

Modifiez un groupe de réplication pour passer au format du moteur

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
       "LogType":"engine-log",
       "LogFormat":"json"
    }'
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
       "LogType":"engine-log",
       "LogFormat":"json"
    }'
```

Modifiez un groupe de réplication pour désactiver la diffusion de journaux du moteur

Pour Linux, macOS ou Unix :

```
aws elasticache modify-replication-group \
    --replication-group-id test-slow-log \
    --apply-immediately \
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "Enabled":false
    }'
```

Pour Windows :

```
aws elasticache modify-replication-group ^
    --replication-group-id test-slow-log ^
    --apply-immediately ^
    --log-delivery-configurations '
    {
      "LogType":"engine-log",
      "Enabled":false
    }'
```

Surveillance de l'utilisation à l'aide de CloudWatch métriques

ElastiCache fournit des métriques qui vous permettent de surveiller vos clusters. Vous pouvez accéder à ces statistiques via CloudWatch. Pour plus d'informations CloudWatch, consultez la CloudWatch documentation.

ElastiCache fournit à la fois des métriques au niveau de l'hôte (par exemple, l'utilisation du processeur) et des métriques spécifiques au logiciel du moteur de cache (par exemple, les mises en cache et les erreurs de cache). Ces métriques sont mesurées et publiées pour chaque nœud de cache toutes les 60 secondes.

> **Important**
>
> Vous devriez envisager de définir des CloudWatch alarmes pour certaines mesures clés, afin d'être averti si les performances de votre cluster de cache commencent à se dégrader. Pour plus d'informations, consultez Quelles métriques dois-je surveiller ? dans ce guide.

Rubriques

- Métriques au niveau de l'hôte
- Métriques pour Valkey et Redis OSS
- Métriques pour Memcached
- Quelles métriques dois-je surveiller ?
- Choix des périodes et des statistiques de métriques
- Surveillance des métriques des CloudWatch clusters et des nœuds

Métriques au niveau de l'hôte

L'espace de noms `AWS/ElastiCache` inclut les métriques au niveau de l'hôte suivantes pour les nœuds de cache individuels. Ces métriques sont mesurées et publiées pour chaque nœud de cache toutes les 60 secondes.

Voir aussi

- Métriques pour Valkey et Redis OSS

Métrique	Description	Unit
`CPUUtilization`	Le pourcentage d'utilisation de l'UC pour l'hôte entier. Valkey et Redis OSS étant à thread unique, nous vous recommandons de surveille	Pourcentage

Métrique	Description	Unit
	r les `EngineCPUUtilization` métriques pour les nœuds contenant 4 v. ou plus. CPUs	
`CPUCreditBalance`	Nombre de crédits UC gagnés qu'une instance a accumulés depuis son lancement ou son démarrage. Pour T2 Standard, le CPUCredit solde inclut également le nombre de crédits de lancement accumulés. Les crédits sont accumulés dans le solde de crédits quand ils sont gagnés et supprimés du solde de crédits lorsqu'ils sont dépensés. Le solde de crédits présente une limite maximum qui est déterminée par la taille de l'instanc e. Une fois que la limite est atteinte, tous les nouveaux crédits gagnés sont rejetés. Pour les instances T2 Standard, les crédits de lancement ne sont pas comptés dans la limite. Les crédits figurant dans le CPUCredit solde peuvent être dépensés par l'instance pour dépasser son taux d'utilisation du processeur de base. Les métriques de crédits UC sont disponibles uniquement toutes les 5 minutes. Cette métrique n'est pas disponible pour les instances de performance à capacité extensible T2.	Crédits (minutes vCPU)

Métrique	Description	Unit
`CPUCreditUsage`	Nombre de crédits UC dépensés par l'instance pour l'utilisation de l'UC. Un crédit de processeur équivaut à un vCPU fonctionnant à 100 % d'utilisation pendant une minute ou une combinaison équivalente de vCPUs, d'utilisation et de temps (par exemple, un vCPU fonctionnant à 50 % d'utilisation pendant deux minutes ou deux vCPU CPUs fonctionnant à 25 % d'utilisation pendant deux minutes). Les métriques de crédits UC sont disponibles uniquement toutes les 5 minutes. Si vous spécifiez une période supérieure à cinq minutes, utilisez la statistique Sum au lieu de la statistique Average. Cette métrique n'est pas disponible pour les instances de performance à capacité extensible T2.	Crédits (minutes vCPU)
`FreeableMemory`	Espace mémoire disponible sur l'hôte. Ceci est dérivé des tampons de RAM et de cache que le SE affiche 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

Métrique	Description	Unit
`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 mis en file d'attente ou ignorés flottee que la bande passante agrégée entrante a dépassé le maximum de l'instance.	Nombre
`NetworkConntrackAllowanceExceeded`	Nombre de paquets ignorés flottee que le suivi des connexions a dépassé le maximum de l'instance et que de nouvelles connexions n'ont pas pu être établies. Cela peut entraîner une perte de paquets pour le trafic vers ou en provenance de l'instance.	Nombre
`NetworkBandwidthOutAllowanceExceeded`	Nombre de paquets mis en file d'attente ou ignorés flottee que la bande passante agrégée sortante a dépassé le maximum de l'instance.	Nombre
`NetworkPacketsPerSecondAllowanceExceeded`	Le nombre de paquets mis en file d'attente ou abandonnés parce que les paquets bidirectionnels par seconde ont dépassé le maximum pour 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 rafale par minute.	Nombre
`NetworkMaxPacketsOut`	Nombre maximal de paquets transmis par seconde par minute.	Nombre

Métrique	Description	Unit
`SwapUsage`	Nombre de permutations utilisées sur l'hôte.	Octets

Métriques pour Valkey et Redis OSS

L'espace de `Amazon ElastiCache` noms inclut les métriques Valkey et Redis OSS suivantes. Ces métriques sont les mêmes lorsque vous utilisez le moteur Valkey.

À l'exception de`ReplicationLag`, `EngineCPUUtilizationSuccessfulWriteRequestLatency`, et`SuccessfulReadRequestLatency`, ces métriques sont dérivées de la info commande. Chaque métrique est calculée au niveau du nœud de cache.

Pour une documentation complète de la info commande, consultez http://valkey. io/commands/info.

Voir aussi

- Métriques au niveau de l'hôte

Métrique	Description	Unit
`ActiveDefragHits`	Nombre de réallocations de valeur par minute effectuées par le processus de défragmentation actif. Ceci est dérivé des `active_defrag_hits` statistiques d'INFO.	Nombre
`AuthenticationFailures`	Nombre total de tentatives infructueuses d'authentification auprès de Valkey ou Redis OSS à l'aide de la commande AUTH. Vous trouverez plus d'informations sur les échecs d'authentification individuels à l'aide de la commande ACL LOG. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre

Métrique	Description	Unit
`BytesUsedForCache`	Le nombre total d'octets alloués par Valkey ou Redis OSS à toutes fins utiles, y compris le jeu de données, les tampons, etc.	Octets
	`Dimension: Tier=Memory` pour les clusters Valkey ou Redis OSS en utilisant Hiérarchisation des données ElastiCache : Le nombre total d'octets utilisés pour le cache par la mémoire. Il s'agit de la valeur de la `used_memory` statistique sur INFO.	Octets
	`Dimension: Tier=SSD` pour les clusters Valkey ou Redis OSS en utilisant Hiérarchisation des données ElastiCache : Le nombre total d'octets utilisés pour le cache par SSD.	Octets
`BytesReadFromDisk`	Nombre total d'octets lus sur le disque par minute. Pris en charge uniquement pour les clusters utilisant Hiérarchisation des données ElastiCache.	Octets
`BytesWrittenToDisk`	Nombre total d''octets écrits sur le disque par minute. Pris en charge uniquement pour les clusters utilisant Hiérarchisation des données ElastiCache.	Octets
`CacheHits`	Le nombre de recherches réussies de clés en lecture seule dans le dictionnaire principal. Ceci est dérivé des `keyspace_hits` statistiques d'INFO.	Nombre
`CacheMisses`	Le nombre de recherches non-réussies de clés en lecture seule dans le dictionnaire principal . Ceci est dérivé des `keyspace_misses` statistiques d'INFO.	Nombre

Métrique	Description	Unit
`CommandAuthorizationFailures`	Nombre total de tentatives infructueuses par les utilisateurs d'exécuter des commandes qu'ils n'ont pas l'autorisation d'appeler. Vous trouverez plus d'informations sur les échecs d'authentification individuels à l'aide de la commande ACL LOG. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
`CacheHitRate`	Indique l'efficacité d'utilisation de l'instance Valkey ou Redis OSS. Si le ratio de cache est inférieur à ~0,8, cela indique qu'une quantité importante de clés est expulsée, expirée ou n'existe pas. Ce ratio est calculé en utilisant les statistiques `cache_hits` et `cache_misses` de la manière suivante : `cache_hits /(cache_hits + cache_misses)` .	Pourcentage
`ChannelAuthorizationFailures`	Nombre total de tentatives infructueuses par les utilisateurs d'accéder aux canaux auxquels ils n'ont pas l'autorisation d'accéder. Vous trouverez plus d'informations sur les échecs d'authentification individuels à l'aide de la commande ACL LOG. Nous vous suggérons de déclencher une alarme sur cette métrique pour détecter les tentatives d'accès non autorisés.	Nombre
`CurrConnections`	Le nombre de connexions client, à l'exclusion des connexions issues des répliques de lecture. ElastiCache utilise 4 à 6 connexions pour surveiller le cluster dans chaque cas. Ceci est dérivé des `connected_clients` statistiques d'INFO.	Nombre

Métrique	Description	Unit
`CurrItems`	Nombre d'éléments dans le cache. Ceci est dérivé de la `keyspace` statistique, en additionnant toutes les touches de l'ensemble du keyspace.	Nombre
	`Dimension: Tier=Memory` pour les clusters utilisant Hiérarchisation des données ElastiCache. Nombre d'éléments en mémoire.	Nombre
	`Dimension: Tier=SSD` (lecteur à état solide) pour les clusters utilisant Hiérarchisation des données ElastiCache. Nombre d'éléments dans le disque SSD.	Nombre
`CurrVolatileItems`	Nombre total de clés dans toutes les bases de données qui ont un ttl défini. Ceci est dérivé de la `expires` statistique, en additionnant toutes les touches avec un ttl défini dans l'ensemble du keyspace.	Nombre
`DatabaseCapacityUsagePercentage`	Pourcentage de la capacité de données totale pour le cluster en cours d'utilisation. Sur les instances Data Tiered, la métrique est calculée sous la forme `(used_memory - mem_not_counted_for_evict + SSD used) / (maxmemory + SSD total capacity)` , où `used_memory` et `maxmemory` est extraite de INFO. Dans tous les autres cas, la métrique est calculée à l'aide de `used_memory/maxmemory` .	Pourcentage

Métrique	Description	Unit

`DatabaseCapacityUsageCountedForEvictPercentage`	Pourcentage de la capacité de données totale pour le cluster en cours d'utilisation, à l'exclusion de la mémoire utilisée pour le traitement et le COB. Cette métrique est calculée comme suit :
	
	`used_memory - mem_not_counted_for_evict/maxmemory`
	
	Sur les instances avec hiérarchisation des données, la métrique est calculée comme suit :
	
	`(used_memory + SSD used) / (maxmemory + SSD total capacity)`
	
	où `used_memory` et `maxmemory` sont tirés d'INFO
		Pourcentage

`DatabaseMemoryUsagePercentage`	Pourcentage de la mémoire pour le cluster qui est en cours d'utilisation. Ceci est calculé à `used_memory/maxmemory` partir de INFO.	Pourcentage
`DatabaseMemoryUsageCountedForEvictPercentage`	Pourcentage de la mémoire pour le cluster qui est en cours d'utilisation, à l'exclusion de la mémoire utilisée pour le traitement et le COB. Ceci est calculé à `used_memory-mem_not_counted_for_evict/maxmemory` partir de INFO.	Pourcentage

Métrique	Description	Unit
`DB0AverageTTL`	Expose `avg_ttl` le DBO à partir des `keyspace` statistiques de la commande INFO. Les réplicas ne font pas expirer les clés, mais attendent plutôt que les nœuds primaires fassent expirer les clés. Lorsqu'un nœud primaire fait expirer une clé (ou l'expulse en tant que moins récemment utilisée), il synthétise une commande `DEL`, qui est transmise à tous les réplicas. Par conséquent, DB0AverageTTL a pour valeur 0 pour les nœuds de réplica, car ils ne font pas expirer les clés et ne suivent donc pas la TTL.	Millisecondes

Métrique	Description	Unit
`EngineCPUUtilization`	Fournit l'utilisation du processeur du thread du moteur Valkey ou Redis OSS. Valkey et Redis OSS étant mono-thread, vous pouvez utiliser cette métrique pour analyser la charge du processus lui-même. La `EngineCPU Utilization` métrique fournit une visibilit é plus précise du processus. La métrique `CPUUtilization` complète la métrique `CPUUtilization` préexistante qui expose l'utilisation du CPU pour l'instance du serveur dans son ensemble, y compris les autres processus du système d'exploitation et les autres processus de gestion. Pour les types de nœuds plus grands avec quatre v CPUs ou plus, utilisez la `EngineCPUUtilization` métrique pour surveiller et définir des seuils de dimensionnement. **Note** Sur un ElastiCache hôte, des processus d'arrière-plan surveillent l'hôte afin de fournir une expérience de base de données gérée. Ces processus en arrière-plan peuvent prendre une part importante de la charge de travail du processeur. Cela n'est pas significa tif sur les hôtes plus grands avec plus de deux CPUs v. Mais cela peut affecter les hôtes plus petits avec 2 V CPUs ou moins. Si vous ne surveille z que la `EngineCPUUtilizati on` métrique, vous ne serez pas au courant des situations dans lesquelles l'hôte est surchargé en raison à la fois	Pourcentage

Métrique	Description	Unit
	d'une utilisation élevée du processeur par Valkey ou Redis OSS et d'une utilisation élevée du processeur due aux processus de surveillance en arrière-plan. Par conséquent, nous vous recommandons de surveiller la `CPUUtilization` métrique pour les hôtes dont la valeur est inférieure CPUs ou égale à deux v.	
`Evictions`	Nombre de clés qui ont été expulsées en raison de la limite `maxmemory` . Ceci est dérivé des `evicted_keys` statistiques d'INFO.	Nombre
`GlobalDatastoreReplicationLag`	Il s'agit du décalage entre le nœud primaire de la région secondaire et le nœud primaire de la région primaire. Pour Valkey ou Redis OSS activé en mode cluster, le décalage indique le délai maximal entre les partitions.	Secondes
`IamAuthenticationExpirations`	Le nombre total de connexions Valkey ou Redis OSS authentifiées par IAM expirées. Vous trouverez plus d'informations sur Authentification avec IAM dans le guide de l'utilisateur.	Nombre
`IamAuthenticationThrottling`	Le nombre total de demandes Valkey ou Redis OSS AUTH ou HELLO limitées authentifiées par IAM. Vous trouverez plus d'informations sur Authentification avec IAM dans le guide de l'utilisateur.	Nombre
`IsMaster`	Indique si le nœud est le nœud primaire de la partition/du cluster courant. La métrique peut être égale à 0 (non primaire) ou 1 (primaire).	Nombre

Métrique	Description	Unit
`KeyAuthorizationFailures`	Nombre total de tentatives infructueuses par les utilisateurs d'accéder aux clés auxquelles ils n'ont pas l'autorisation d'accéder. Vous trouverez plus d'informations sur les échecs d'authentification individuels à l'aide de la commande ACL LOG. Nous vous suggérons de déclencher une alarme pour détecter les tentatives d'accès non autorisés.	Nombre
`KeysTracked`	Le nombre de clés suivies par le suivi des clés Valkey ou Redis OSS en pourcentage de. `tracking-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
`MemoryFragmentationRatio`	Indique l'efficacité de l'allocation de mémoire du moteur Valkey ou Redis OSS. Certains seuils indiqueront différents comportements. La valeur recommandée est d'avoir une fragmentation supérieure à 1.0. Ceci est calculé à partir `mem_fragmentation_ratio statistic` de l'INFO.	Nombre

Métrique	Description	Unit
`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 d'INFO. **Note** Si vous utilisez ElastiCache Redis OSS version 5 ou inférieure, entre deux et quatre des connexions signalées par cette métrique sont utilisées ElastiCache pour surveiller le cluster. Toutefois, lors de l'utilisation ElastiCache de Redis OSS version 6 ou ultérieure, les connexions utilisées ElastiCache pour surveiller le cluster ne sont pas incluses dans cette métrique.	Nombre
`NumItemsReadFromDisk`	Nombre total d'éléments récupérés à partir du disque par minute. Pris en charge uniquement pour les clusters utilisant Hiérarchisation des données ElastiCache.	Nombre
`NumItemsWrittenToDisk`	Nombre total d'éléments écrits sur disque par minute. Pris en charge uniquement pour les clusters utilisant Hiérarchisation des données ElastiCache.	Nombre
`MasterLinkHealthStatus`	Cet état a deux valeurs : 0 ou 1. La valeur 0 indique que les données du nœud ElastiCache principal ne sont pas synchronisées avec Valkey ou Redis OSS activé. EC2 Une valeur égale à 1 signifie que les données sont synchronisées. Pour terminer la migration, utilisez l'opération CompleteMigrationAPI.	Booléen

Métrique	Description	Unit
`Reclaimed`	Nombre total d'événements d'expiration de clé. Ceci est dérivé des `expired_keys` statistiq ues d'INFO.	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 représent ative de la charge d'écriture sur le groupe de réplication. Ceci est dérivé des `master_re pl_offset` statistiques d'INFO.	Octets
`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'applica tion par le réplica des modifications provenant du nœud principal. Pour Valkey 7.2 et versions ultérieures, et Redis OSS 5.0.6 et versions ultérieures, le décalage peut être mesuré en millisecondes.	Secondes
`SaveInProgress`	Cette métrique binaire renvoie la valeur 1 lorsqu'un enregistrement en arrière-plan (avec ou sans autoclonage) est en cours. Sinon, elle renvoie la valeur 0. Un processus d'enregis trement en arrière-plan est généralement utilisé lors des instantanés et des synchroni sations. Ces opérations peuvent entraîner une dégradation des performances. La métrique `SaveInProgress` vous permet de diagnosti quer si la dégradation des performances a été provoquée par un processus d'enregis trement en arrière-plan. Ceci est dérivé des `rdb_bgsave_in_progress` statistiques d'INFO.	Booléen

Métrique	Description	Unit
`TrafficManagementActive`	Indique si ElastiCache Redis OSS gère activement le trafic en ajustant le trafic alloué aux commandes entrantes, à la surveillance ou à la réplication. Le trafic est géré lorsque le nombre de commandes envoyées au nœud est supérieur à ce qui peut être traité par Valkey ou Redis OSS et est utilisé pour maintenir la stabilité et le fonctionnement optimal du moteur. Tout point de données de valeur 1 peut indiquer que le nœud est sous-dimensionné pour la charge de travail fournie. **Note** Si cette métrique reste active, évaluez le cluster pour décider s'il est nécessaire de l'augmenter ou de le réduire. Les métriques associées incluent `NetworkBandwidthOutAllowanceExceeded` et `EngineCPUUtilization`.	Booléen
`SuccessfulWriteRequestLatency`	Latence des demandes d'écriture réussies. Statistiques valides : moyenne, somme, minimum, maximum, nombre d'échantillons, tout percentile compris entre p0 et p100. Le nombre d'échantillons inclut uniquement les commandes exécutées avec succès.	Microsecondes

Métrique	Description	Unit
`SuccessfulReadRequestLatency`	Latence des demandes de lecture réussies. Statistiques valides : moyenne, somme, minimum, maximum, nombre d'échantillons, tout percentile compris entre p0 et p100. Le nombre d'échantillons inclut uniquement les commandes exécutées avec succès.	Microsecondes
`ErrorCount`	Nombre total de commandes ayant échoué au cours de la période spécifiée. Statistiques valides : moyenne, somme, min, max	Nombre

CPUUtilization Disponibilité du moteur

AWS Les régions répertoriées ci-dessous sont disponibles sur tous les types de nœuds pris en charge.

Région	Nom de la région
`us-east-2`	USA Est (Ohio)
`us-east-1`	USA Est (Virginie du Nord)
`us-west-1`	USA Ouest (Californie du Nord)
`us-west-2`	USA Ouest (Oregon)
`ap-northeast-1`	Asie Pacifique (Tokyo)
`ap-northeast-2`	Asie-Pacifique (Séoul)
`ap-northeast-3`	Asie-Pacifique (Osaka)
`ap-east-1`	Asie-Pacifique (Hong Kong)

Région	Nom de la région
ap-south-1	Asie-Pacifique (Mumbai)
ap-southeast-1	Asie-Pacifique (Singapour)
ap-southeast-2	Asie-Pacifique (Sydney)
ap-southeast-3	Asie-Pacifique (Jakarta)
ca-central-1	Canada (Centre)
cn-north-1	Chine (Beijing)
cn-northwest-2	Chine (Ningxia)
me-south-1	Moyen-Orient (Bahreïn)
eu-central-1	Europe (Francfort)
eu-west-1	Europe (Irlande)
eu-west-2	Europe (Londres)
eu-west-3	UE (Paris)
eu-south-1	Europe (Milan)
af-south-1	Afrique (Le Cap)
eu-north-1	Europe (Stockholm)
sa-east-1	Amérique du Sud (São Paulo)
us-gov-west-1	AWS GovCloud (US-Ouest)
us-gov-east-1	AWS GovCloud (USA Est)

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, le temps CPU total consommé par ces commandes et le CPU moyen consommé

par exécution de commande. Pour chaque type de commande, la ligne suivante est ajoutée : `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

Les mesures de latence répertoriées ci-dessous sont calculées à l'aide des statistiques de commande d'INFO. Elles sont calculées de la manière suivante : `delta(usec)/delta(calls)`. `delta` est calculé comme la diff. dans une minute. La latence est définie comme le temps nécessaire au processeur ElastiCache pour traiter la commande. Notez que pour les clusters utilisant la hiérarchisation des données, le temps nécessaire à la récupération des éléments à partir du SSD n'est pas inclus dans ces mesures.

Pour une liste complète des commandes disponibles, consultez les commandes dans la documentation de Valkey.

Métrique	Description	Unit
`ClusterBasedCmds`	Nombre total de commandes basées sur un cluster. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur un cluster (`cluster slot`,`cluster info`, etc.).	Nombre
`ClusterBasedCmdsLatency`	Latence des commandes basées sur un cluster.	Microsecondes
`EvalBasedCmds`	Nombre total de commandes pour les commandes basées sur eval. Ceci est dérivé de la `commandstats` statistique en additionnanteval,. evalsha	Nombre
`EvalBasedCmdsLatency`	Latence des commandes basées sur Eval.	Microsecondes
`GeoSpatialBasedCmds`	Nombre total de commandes pour les commandes basées sur la géolocalisation. Ceci est dérivé de la `commandstats` statistique. Il est dérivé en additionnant tous les types de commandes géo : geoadd, geodist, geohash, geopos, georadius et georadiusbymember.	Nombre

Métrique	Description	Unit
`GeoSpatialBasedCmdsLatency`	Latence des commandes basées sur la géolocalisation.	Microsecondes
`GetTypeCmds`	Le nombre total de commandes basées sur les types de commandes read-only. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes de read-only type (gethget,scard,lrange,, etc.)	Nombre
`GetTypeCmdsLatency`	Latence des commandes de lecture.	Microsecondes
`HashBasedCmds`	Nombre total de commandes basées sur le hachage. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur un ou plusieurs hachages (hget,,,hkeys, hvalshdel, etc.).	Nombre
`HashBasedCmdsLatency`	Latence des commandes basées sur le hachage.	Microsecondes
`HyperLogLogBasedCmds`	Nombre total de commandes basées sur `HyperLogLog` . Ceci est dérivé de la `commandstats` statistique en additionnant tous les pf types de commandes (pfaddpfcount,pfmerge,, etc.).	Nombre
`HyperLogLogBasedCmdsLatency`	Latence des commandes HyperLogLog basées.	Microsecondes
`JsonBasedCmds`	Nombre total de commandes JSON, y compris les commandes de lecture et d'écriture. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes JSON qui agissent sur les clés JSON.	Nombre

Métrique	Description	Unit
`JsonBasedCmdsLatency`	Latence de toutes les commandes JSON, y compris les commandes de lecture et d'écriture.	Microseconds
`JsonBasedGetCmds`	Nombre total de commandes en lecture seule JSON. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes de lecture JSON qui agissent sur les clés JSON.	Nombre
JsonBasedGetCmdsLatency	Latence des commandes en lecture seule JSON.	Microseconds
JsonBasedSetCmds	Nombre total de commandes d'écriture JSON. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes d'écriture JSON qui agissent sur les clés JSON.	Nombre
JsonBasedSetCmdsLatency	Latence des commandes d'écriture JSON.	Microseconds
`KeyBasedCmds`	Nombre total de commandes basées sur une clé. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur une ou plusieurs touches dans plusieurs structures de données (delexpire,rename,, etc.).	Nombre
`KeyBasedCmdsLatency`	Latence des commandes basées sur les clés.	Microseconds
`ListBasedCmds`	Nombre total de commandes basées sur une liste. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur une ou plusieurs listes (lindex,lrange,lpush,ltrim, etc.).	Nombre

Métrique	Description	Unit
`ListBasedCmdsLatency`	Latence des commandes basées sur la liste.	Microsecondes
NonKeyTypeCmds	Nombre total de commandes non basées sur des clés. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui n'agissent pas sur une touche, par exempleacl, dbsize ou. info	Nombre
NonKeyTypeCmdsLatency	Latence des non-key-based commandes.	Microsecondes
`PubSubBasedCmds`	Nombre total de commandes pour les pub/sub fonctionnalités. Ceci est dérivé des `commandstats` statistiques en additionnant toutes les commandes utilisées pour les pub/sub fonctionnalités : psubscribepublish,pubsub,punsubscribe,ssubscribe,sunsubscribe, spublishsubscribe, etunsubscribe.	Nombre
`PubSubBasedCmdsLatency`	Latence des commandes basées sur pub/sub.	Microsecondes
`SetBasedCmds`	Nombre total de commandes basées sur un ensemble. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur un ou plusieurs ensembles (scard,sdiff,sadd,sunion, etc.).	Nombre
`SetBasedCmdsLatency`	Latence des commandes basées sur des ensembles.	Microsecondes

Métrique	Description	Unit
`SetTypeCmds`	Le nombre total de commandes de type write. Ceci est dérivé de la `commandstats` statistiq ue en additionnant tous les mutative types de commandes qui opèrent sur les données (sethset,sadd,lpop,, etc.)	Nombre
`SetTypeCmdsLatency`	Latence des commandes d'écriture.	Microseco ndes
`SortedSetBasedCmds`	Nombre total de commandes qui sont triées en fonction d'un ensemble. Ceci est dérivé de la `commandstats` statistique en additionnant toutes les commandes qui agissent sur un ou plusieurs ensembles triés (zcount,zrange,zrank,zadd, etc.).	Nombre
`SortedSetBa sedCmdsLatency`	Latence des commandes basées sur le tri.	Microseco ndes
`StringBasedCmds`	Nombre total de commandes basées sur une chaîne. Ceci est dérivé de la `commandst ats` statistique en additionnant toutes les commandes qui agissent sur une ou plusieurs chaînes (strlen,setex,setrange, etc.).	Nombre
`StringBased CmdsLatency`	Latence des commandes basées sur des chaînes	Microseco ndes
`StreamBasedCmds`	Nombre total de commandes basées sur un flux. Ceci est dérivé de la `commandst ats` statistique en additionnant toutes les commandes qui agissent sur un ou plusieurs types de données de flux (xrange,xlen,xadd,xdel, etc.).	Nombre

Métrique	Description	Unit
StreamBasedCmdsLatency	Latence des commandes basées sur des flux.	Microsecondes

Métriques pour Memcached

L'espace de noms `AWS/ElastiCache` inclut les métriques Memcached suivantes.

L'espace de ElastiCache noms AWS/inclut les métriques suivantes qui sont dérivées de la commande Memcached stats. Chaque métrique est calculée au niveau du nœud de cache.

Voir aussi

- Métriques au niveau de l'hôte

Métrique	Description	Unit
BytesReadIntoMemcached	Nombre d'octets lus par le nœud de cache à partir du réseau.	Octets
BytesUsedForCacheItems	Nombre d'octets utilisés pour stocker les éléments de cache.	Octets
BytesWrittenOutFromMemcached	Nombre d'octets écrits par le nœud de cache sur le réseau.	Octets
CasBadval	Nombre de requêtes CAS (vérification et définition) reçues par le cache dans lesquelles la valeur CAS ne correspondait pas à la valeur CAS stockée.	Nombre
CasHits	Nombre de requêtes CAS reçues par le cache dans lesquelles la clé demandée a été trouvée et la valeur CAS correspondait.	Nombre

Métrique	Description	Unit
`CasMisses`	Nombre de requêtes CAS reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
`CmdFlush`	Nombre de commandes flush reçues par le cache.	Nombre
`CmdGet`	Nombre de commandes get reçues par le cache.	Nombre
`CmdSet`	Nombre de commandes set reçues par le cache.	Nombre
`CurrConnections`	Nombre de connexions connectées au cache à un instant donné. ElastiCache utilise 2 à 3 connexions pour surveiller le cluster. En plus de ce qui précède, Memcached crée un nombre de connexions internes égal à deux fois celui des threads utilisés pour ce type de nœud. Le nombre de threads pour les différents types de nœuds figure dans le paramètre `Nodetype Specific Parameters` du groupe de paramètres applicable. Le nombre total de connexions est la somme des connexions client, des connexions à superviser et des connexions internes mentionnées ci-dessus.	Nombre
`CurrItems`	Nombre d'éléments actuellement stockés dans le cache.	Nombre
`DecrHits`	Nombre de requêtes decrement reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre

Métrique	Description	Unit
`DecrMisses`	Nombre de requêtes decrement reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
`DeleteHits`	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
`DeleteMisses`	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
`Evictions`	Nombre d'éléments non arrivés à expiration que le cache a expulsés pour libérer de l'espace pour les nouvelles écritures.	Nombre
`GetHits`	Nombre de requêtes get reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
`GetMisses`	Nombre de requêtes get reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
`IncrHits`	Nombre de requêtes increment reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
`IncrMisses`	Nombre de requêtes increment reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
`Reclaimed`	Nombre d'éléments arrivés à expiration que le cache a expulsés pour libérer de l'espace pour les nouvelles écritures.	Nombre

Pour Memcached 1.4.14, les métriques supplémentaires suivantes sont fournies.

Métrique	Description	Unit
`BytesUsedForHash`	Nombre d'octets actuellement utilisés par les tables de hachage.	Octets
`CmdConfigGet`	Nombre cumulé de demandes config get.	Nombre
`CmdConfigSet`	Nombre cumulé de demandes config set.	Nombre
`CmdTouch`	Nombre cumulé de demandes touch.	Nombre
`CurrConfig`	Nombre actuel de configurations stockées.	Nombre
`EvictedUnfetched`	Nombre d'éléments valides expulsés du cache récemment utilisé qui n'ont jamais été touchés après leur définition.	Nombre
`ExpiredUnfetched`	Nombre d'éléments arrivés à expiration récupérés du cache récemment utilisé qui n'ont jamais été touchés après leur définition.	Nombre
`SlabsMoved`	Nombre total de pages de la section qui ont été déplacées.	Nombre
`TouchHits`	Nombre de clés qui ont été touchées et auxquelles une nouvelle date d'expiration a été affectée.	Nombre
`TouchMisses`	Nombre d'éléments qui ont été touchés, mais sont introuvables.	Nombre

L'espace de ElastiCache noms AWS/inclut les métriques calculées suivantes au niveau du cache.

Métrique	Description	Unit
`NewConnections`	Nombre de nouvelles connexions reçues par le cache. Celui-ci est dérivé de la statistique Memcached total_connections en enregistrant	Nombre

Métrique	Description	Unit
	la modification de total_connections sur une période de temps. Ce sera toujours au moins 1, en raison d'une connexion réservée à un ElastiCache.	
`NewItems`	Nombre de nouveaux éléments stockés par le cache. Celui-ci est dérivé de la statistique Memcached total_items en enregistrant la modification de total_items sur une période de temps.	Nombre
`UnusedMemory`	Quantité de mémoire non utilisée par les données. Celle-ci est dérivée des statistiq ues Memcached limit_maxbytes et bytes en soustrayant bytes de limit_maxbytes. Étant donné que la surcharge de Memcached utilise de la mémoire en plus de celle utilisée par les données, elle ne UnusedMemory doit pas être considérée comme la quantité de mémoire disponible pour des données supplémentaires. Vous pouvez être confronté à des expulsions, même si vous disposez encore de mémoire inutilisée. Pour de plus amples informations, veuillez consulter Memcached item memory usage.	Octets

Quelles métriques dois-je surveiller ?

Les CloudWatch indicateurs suivants offrent un bon aperçu ElastiCache des performances. 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
- Moteur CPUUtilization
- SwapUsage (Valkey et Redis OSS)
- Evictions
- CurrConnections
- Mémoire (Valkey et Redis OSS)
- Réseau
- Latence
- Réplication
- Gestion du trafic (Valkey et Redis OSS)

CPUUtilization

Il s'agit d'une métrique au niveau de l'hôte représentée en pourcentage. Pour de plus amples informations, veuillez consulter Métriques au niveau de l'hôte.

Valkey et Redis OSS

Pour les types de nœuds plus petits avec 2 V CPUs ou moins, utilisez la `CPUUtilization` métrique pour surveiller votre charge de travail.

En général, nous vous suggérons de définir votre seuil à 90 % de votre UC disponible. Valkey et Redis OSS étant tous deux à thread unique, 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 %.

Vous devez déterminer votre propre seuil, en fonction du nombre de cœurs dans le nœud de cache que vous utilisez. Si vous dépasserez ce seuil, et que les demandes de lecture constituent votre

principale charge de travail, ajustez votre cluster de cache en ajoutant des réplicas en lecture. Si la principale charge de travail provient de demandes d'écriture, selon la configuration de votre cluster, nous vous recommandons de :

- Clusters Valkey ou Redis OSS (mode cluster désactivé) : augmentez votre capacité en utilisant un type d'instance de cache plus important.
- Clusters Valkey ou Redis OSS (mode cluster activé) : ajoutez des partitions supplémentaires pour répartir la charge d'écriture sur un plus grand nombre de nœuds principaux.

> **Tip**
>
> Au lieu d'utiliser la métrique au niveau de l'hôte`CPUUtilization`, les utilisateurs de Valkey et Redis OSS peuvent utiliser la métrique`EngineCPUUtilization`, qui indique le pourcentage d'utilisation sur le cœur du moteur Valkey ou Redis OSS. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for Valkey and Redis OSS.

Pour les types de nœuds plus grands avec 4 V CPUs ou plus, vous pouvez utiliser la `EngineCPUUtilization` métrique, qui indique le pourcentage d'utilisation sur le cœur du moteur Valkey ou Redis OSS. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for Redis OSS.

Memcached

Puisque Memcached est multi-thread, cette métrique peut atteindre jusqu'à 90 %. Si vous dépassez ce seuil, augmentez votre cluster de cache en utilisant un type de nœud de cache plus important ou augmentez la taille en ajoutant d'autres nœuds de cache.

Moteur CPUUtilization

Pour les types de nœuds plus grands avec 4 V CPUs ou plus, vous pouvez utiliser la `EngineCPUUtilization` métrique, qui indique le pourcentage d'utilisation sur le cœur du moteur Redis OSS. Pour savoir si cette métrique est disponible sur vos nœuds et pour plus d'informations, consultez Metrics for Valkey and Redis OSS.

Pour plus d'informations, consultez la CPUssection consacrée à la surveillance des meilleures pratiques avec Amazon ElastiCache pour Redis OSS à l'aide d'Amazon CloudWatch.

SwapUsage (Valkey et Redis OSS)

Il s'agit d'une métrique au niveau de l'hôte, publiée en octets. Pour de plus amples informations, veuillez consulter Métriques au niveau de l'hôte.

Si la `FreeableMemory` CloudWatch métrique est proche de 0 (c'est-à-dire inférieure à 100 Mo) ou supérieure à la `SwapUsage` `FreeableMemory` métrique, cela indique qu'un nœud est soumis à une pression de mémoire. Si cela se produit, consultez les rubriques suivantes :

- S'assurer que vous disposez de suffisamment de mémoire pour créer un instantané Valkey ou Redis OSS
- Gestion de la mémoire réservée pour Valkey et Redis OSS

Evictions

Il s'agit d'une métrique de moteur de cache. Nous vous recommandons de choisir votre propre seuil d'alarme pour cette métrique en fonction des besoins de votre application.

Si vous utilisez Memcached et que vous dépassez le seuil que vous avez choisi, augmentez votre cluster en utilisant un type de nœud plus grand ou augmentez la taille en ajoutant de nouveaux nœuds.

CurrConnections

Il s'agit d'une métrique de moteur de cache. 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.

Pour plus d'informations, consultez la section Connexions sur Surveillance des meilleures pratiques avec Amazon ElastiCache pour Redis OSS à l'aide d'Amazon CloudWatch.

Mémoire (Valkey et Redis OSS)

La mémoire est un aspect essentiel de Valkey et Redis OSS. 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 commande INFO.

Pour plus d'informations, consultez la section Mémoire de la section Surveillance des meilleures pratiques avec Amazon ElastiCache pour Redis OSS à l'aide d'Amazon CloudWatch.

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 les ElastiCache tarifs Amazon.

Pour plus d'informations, consultez la section Réseau sur la section Surveillance des meilleures pratiques avec Amazon ElastiCache pour Redis OSS à l'aide d'Amazon CloudWatch.

Latence

La mesure du temps de réponse ElastiCache pour une instance for Valkey peut être abordée de différentes manières en fonction du niveau de granularité requis. Les étapes clés qui contribuent au temps de réponse global côté serveur ElastiCache pour Valkey sont le prétraitement des commandes, l'exécution des commandes et le post-traitement des commandes.

Les métriques de latence spécifiques à la commande dérivées de la commande Valkey INFO, telles que GetTypeCmdsLatency les SetTypeCmdsLatency métriques, se concentrent spécifiquement sur l'exécution de la logique de commande de base pour la commande Valkey. Ces mesures seront utiles si votre cas d'utilisation consiste à déterminer le temps d'exécution des commandes ou les latences agrégées par structure de données.

Les métriques `SuccessfulWriteRequestLatency` de latence `SuccessfulReadRequestLatency` mesurent le temps total nécessaire au moteur ElastiCache for Valkey pour répondre à une demande.

> **Note**
>
> Des valeurs `SuccessfulWriteRequestLatency` et des `SuccessfulReadRequestLatency` métriques gonflées peuvent se produire lors de l'utilisation du pipeline Valkey avec CLIENT REPLY activé sur le client Valkey. Le pipeline Valkey est une technique permettant d'améliorer les performances en émettant plusieurs commandes à la fois, sans attendre la réponse à chaque commande individuelle. Pour éviter les valeurs exagérées, nous vous recommandons de configurer votre client Valkey pour qu'il achemine les commandes avec CLIENT REPLY OFF.

Pour plus d'informations, consultez la section Latence de la section Surveillance des meilleures pratiques avec Amazon à ElastiCache l'aide d'Amazon CloudWatch.

Réplication

Le volume de données en cours de réplication est visible via le métrique `ReplicationBytes`. Bien que cette métrique soit représentative de la charge d'écriture sur le groupe de réplication, elle ne fournit pas d'informations sur l'intégrité de la réplication. Pour ce faire, vous pouvez utiliser la métrique `ReplicationLag`.

Pour plus d'informations, consultez la section Réplication sur Surveillance des meilleures pratiques avec Amazon ElastiCache pour Redis OSS à l'aide d'Amazon CloudWatch.

Gestion du trafic (Valkey et Redis OSS)

ElastiCache for Redis OSS gère automatiquement le trafic vers un nœud lorsque le nombre de commandes entrantes envoyées au nœud est supérieur à ce qui peut être traité par Valkey ou Redis OSS. Cela vise à maintenir un fonctionnement et une stabilité optimaux du moteur.

Lorsque le trafic est géré activement sur un nœud, la métrique `TrafficManagementActive` émet des points de données de valeur 1. Cela indique que le nœud est peut-être sous-dimensionné pour la charge de travail fournie. Si cette métrique reste à 1 sur de longues périodes, évaluez le cluster pour décider s'il est nécessaire de procéder à une augmentation ou à une montée en puissance.

Pour en savoir plus, consultez la métrique `TrafficManagementActive` sur la page Métriques.

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 ElastiCache exemples sont publiés pendant 60 secondes pour chaque nœud de cache individuel. Pour toute période de 60 secondes, une métrique de nœud de cache ne contiendra qu'un seul échantillon.

Pour plus d'informations sur la façon de récupérer des métriques pour vos nœuds de cache, consultez Surveillance des métriques des CloudWatch clusters et des nœuds.

Surveillance des métriques des CloudWatch clusters et des nœuds

ElastiCache et CloudWatch sont intégrés afin que vous puissiez recueillir une variété de mesures. 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 de cache au cours de la dernière heure.

> **Note**
>
> Les valeurs `StartTime` et `EndTime` fournies dans les exemples ci-dessous sont communiquées à des fins d'illustration. Vous devez les remplacer par les valeurs adaptées à nœuds de cache.

Pour plus d'informations sur ElastiCache les limites, consultez la section Limites de AWS service pour ElastiCache.

Surveillance des métriques des CloudWatch clusters et des nœuds (console)

Pour collecter les statistiques d'utilisation de l'UC d'un cluster de cache

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/elasticache/.
2. Sélectionnez les nœuds de cache dont vous voulez afficher 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 cache de la console de AWS gestion, cliquez sur le nom d'un ou de plusieurs clusters de cache.

      La page détaillée du cluster de cache s'affiche.

   b. Cliquez sur l'onglet Nodes en haut de la fenêtre.

   c. Sur l'onglet Nodes de la fenêtre des détails, sélectionnez les nœuds de cache dont vous souhaitez afficher les métriques.

      La liste des CloudWatch métriques disponibles apparaît en bas de la fenêtre de console.

   d. Cliquez sur la métrique CPU Utilization.

      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 métriques des CloudWatch clusters et des nœuds à l'aide de la CloudWatch CLI

Pour collecter les statistiques d'utilisation de l'UC d'un cluster de cache

- Pour Linux, macOS ou Unix :

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/ElastiCache \
```

```
    --metric-name CPUUtilization \
    --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
    --statistics=Average \
    --start-time 2018-07-05T00:00:00 \
    --end-time 2018-07-06T00:00:00 \
    --period=3600
```

Pour Windows :

```
aws cloudwatch get-metric-statistics ^
    --namespace AWS/ElastiCache ^
    --metric-name CPUUtilization ^
    --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' ^
    --statistics=Average ^
    --start-time 2018-07-05T00:00:00 ^
    --end-time 2018-07-06T00:00:00 ^
    --period=3600
```

Surveillance des métriques des CloudWatch clusters et des nœuds à l'aide de l' CloudWatch API

Pour collecter les statistiques d'utilisation de l'UC d'un cluster de cache

- Appelez l' 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/ElastiCache`
  - `StartTime=2013-07-05T00:00:00`
  - `EndTime=2013-07-06T00:00:00`
  - `Period=60`
  - `MeasureName=CPUUtilization`
  - `Dimensions=CacheClusterId=mycachecluster,CacheNodeId=0002`

Example

```
http://monitoring.amazonaws.com/
    ?Action=GetMetricStatistics
    &SignatureVersion=4
    &Version=2014-12-01
    &StartTime=2018-07-05T00:00:00
    &EndTime=2018-07-06T23:59:00
    &Period=3600
    &Statistics.member.1=Average
    &Dimensions.member.1="CacheClusterId=mycachecluster"
    &Dimensions.member.2="CacheNodeId=0002"
    &Namespace=&AWS;/ElastiCache
    &MeasureName=CPUUtilization
    &Timestamp=2018-07-07T17%3A48%3A21.746Z
    &AWS;AccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Quotas pour ElastiCache

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 consulter les quotas pour ElastiCache, ouvrez la console Service Quotas. Dans le panneau de navigation, sélectionnez Services AWS , puis ElastiCache.

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 relatifs à ElastiCache.

Nom	Valeur par défaut	Description	Nom de la métrique
Nœuds par région	300	Nombre maximal de nœuds sur tous les clusters d'une région. Ce quota s'appliqu e à vos nœuds non réservés dans la région donnée. Vous pouvez avoir jusqu'à 300 nœuds non réservés dans la même région.	NodesPerRegion
Nœuds par cluster (Memcached)	60	Nombre maximal de nœuds d'un cluster Memcached spécifiqu e.	Memcached NodesPerCluster
Nœuds par cluster (mode cluster activé)	90	Le nombre maximum de nœuds dans un cluster Redis OSS ou Valkey individuel.	NodesPerCluster

Nom	Valeur par défaut	Description	Nom de la métrique
Groupes de paramètres par région	300	Nombre maximal de groupes de paramètres que vous pouvez créer dans une région.	ParameterGroup
Groupes de sous-réseaux par région	300	Nombre maximal de groupes de sous-réseaux que vous pouvez créer dans une région.	SubnetGroup
Sous-réseaux par groupe de sous-réseaux	20	Nombre maximal de sous-réseaux que vous pouvez définir pour un groupe de sous-réseaux.	SubnetsPerSubnetGroup
Caches sans serveur par région	40	Le nombre maximum de caches sans serveur dans une région.	ServerlessCache
Instantanés sans serveur par jour et par cache	24	Nombre maximal de snapshots sans serveur que vous pouvez prendre par jour pour chaque cache sans serveur.	SnapshotsPerDayPerCache
Utilisateurs par région	2000	Le nombre total maximum d'utilisateurs que vous pouvez créer dans une région.	Utilisateur

Nom	Valeur par défaut	Description	Nom de la métrique
Groupes d'utilisateurs par région	200	Le nombre maximum de groupes d'utilisa teurs que vous pouvez créer dans une région.	UserGroup
Utilisateurs par groupe d'utilisateurs	100	Le nombre maximum d'utilisateurs que vous pouvez définir pour un groupe d'utilisateurs.	UsersPerUserGroup

Référence

Les rubriques de cette section traitent de l'utilisation de l' ElastiCache API Amazon et de la ElastiCache section du AWS CLI. Sont également présentés les messages d'erreur et les notifications de service courants.

- Utilisation de l' ElastiCache API
- ElastiCache API Reference
- ElastiCache section de la AWS CLI référence
- Messages ElastiCache d'erreur Amazon
- Notifications

Utilisation de l' ElastiCache API

Cette section fournit des descriptions axées sur les tâches relatives à l'utilisation et à la mise en œuvre ElastiCache des opérations. Pour une description complète de ces opérations, consultez le Amazon ElastiCache API Reference.

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 commençant à 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. Cela suppose que vous êtes inscrit AWS et que vous disposez d'un identifiant 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'identifiant de la clé d'accès (l'identifiant de la clé d'accès secrète utilisée) à. AWS
4. AWS utilise l'ID de clé d'accès pour rechercher la clé d'accès secrète.
5. AWS génère une signature à partir des données de la demande et de la clé d'accès secrète en utilisant le même algorithme que celui utilisé pour calculer la signature dans la demande.
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 du POST (lorsque Content-Type est x-www-form-urlencoded application/).

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 HTTPRequest URI est le composant du chemin HTTP absolu de l'URI jusqu'à la chaîne de requête, mais non comprise. Si l' HTTPRequestURI est vide, utilisez une barre oblique (/).

3. Calculez un HMAC conforme à la RFC 2104 avec la chaîne que vous venez de créer, votre clé d'accès secrète comme clé et/ou SHA1 comme algorithme de SHA256 hachage.

   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://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
```

Pour la chaîne de requête précédente, vous devez calculer la signature HMAC sur la chaîne suivante.

```
GET\n
    elasticache.amazonaws.com\n
    Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache
%2Faws4_request
    &X-Amz-Date=20141201T223649Z
    &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-
amz-date
        content-type:
        host:elasticache.us-west-2.amazonaws.com
        user-agent:CacheServicesAPICommand_Client
    x-amz-content-sha256:
    x-amz-date:
```

Le résultat est la demande signée suivante.

```
https://elasticache.us-west-2.amazonaws.com/
    ?Action=DescribeCacheClusters
    &CacheClusterIdentifier=myCacheCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2014-12-01
    &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
```

```
    &X-Amz-Date=20141201T223649Z
    &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 rubrique processus de signature Signature Version 4 et ses sous-rubriques.

Bibliothèques disponibles

AWS fournit des kits de développement logiciel (SDKs) aux développeurs de logiciels qui préfèrent créer des applications en utilisant un langage spécifique APIs plutôt que l'API Query. Ils SDKs fournissent des fonctions de base (non incluses dans le APIs), telles que l'authentification des demandes, les nouvelles tentatives de demande et la gestion des erreurs, afin de faciliter le démarrage. SDKs 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, consultez Exemples de code et bibliothèques.

Applications de dépannage

ElastiCache fournit des erreurs spécifiques et descriptives pour vous aider à résoudre les problèmes lors de l'interaction avec l' ElastiCache API.

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 savoir si une erreur s'est produite est de rechercher un `Error` nœud dans la réponse de l' ElastiCache API.

XPath la syntaxe fournit un moyen simple de rechercher la présence d'un `Error` nœud, ainsi qu'un moyen facile de récupérer le code d'erreur et le message. L'extrait de code suivant utilise Perl et le

XPath module XML : : pour déterminer si une erreur s'est produite lors d'une requête. 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 vous conseillons d'utiliser les processus suivants pour diagnostiquer et résoudre les problèmes avec l'API ElastiCache .

- Vérifiez qu' ElastiCache il fonctionne correctement.

  Pour ce faire, il suffit d'ouvrir une fenêtre de navigateur et de soumettre une demande de requête au ElastiCache service (par exemple https://elasticache.amazonaws.com). Une erreur interne du serveur MissingAuthenticationTokenException ou 500 confirme que le service est disponible et qu'il répond aux demandes.
- Vérifiez la structure de votre demande.

  Chaque ElastiCache opération possède une page de référence dans le Guide de référence de ElastiCache l'API. 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.

  ElastiCache dispose d'un forum de discussion où vous pouvez rechercher des solutions aux problèmes que d'autres ont rencontrés en cours de route. Pour consulter le forum, rendez-vous à l'adresse

  https://forums.aws.amazon.com/ .

Configuration de l'interface de ligne de ElastiCache commande

Cette section décrit les conditions requises pour exécuter les outils de ligne de commande, explique où trouver ces outils, comment définir les outils et leur environnement, et propose différents exemples courants d'utilisation.

Suivez les instructions de cette rubrique uniquement si vous utilisez le AWS CLI formulaire ElastiCache.

> **Important**
>
> L'interface de ligne de ElastiCache commande (CLI) Amazon ne prend en charge aucune ElastiCache amélioration après la version 2014-09-30 de l'API. Pour utiliser les nouvelles ElastiCache fonctionnalités de la ligne de commande, utilisez l'interface de ligne de AWS commande.

Rubriques

- Prérequis
- Téléchargement des outils de ligne de commande
- Installation des outils
- Indication des informations d'identification pour les outils
- Variables d'environnement

Prérequis

Ce document part du principe que vous pouvez travailler dans un environnement Linux/UNIX ou Windows. Les outils de ligne de ElastiCache commande Amazon fonctionnent également sur Mac OS X, qui est un environnement basé sur Unix ; toutefois, aucune instruction spécifique pour Mac OS X n'est incluse dans ce guide.

En guise de convention, tout le texte de ligne de commande est préfixé d'une invite de ligne de commande **`PROMPT>`** générique. Il est probable que la demande de ligne de commande de votre machine soit différente. Nous l'utilisons également **`$`** pour indiquer une commande Linux/UNIX spécifique et **`C:\>`** pour une commande spécifique à Windows. L'exemple de sortie résultant de la commande s'affiche immédiatement après, sans préfixe.

Java Runtime Environment

Les outils de ligne de commande utilisés dans ce manuel nécessitent l'exécution de Java version 5 ou suivante. Une installation JRE ou JDK est acceptable. Pour consulter et télécharger JREs pour diverses plateformes, y compris Linux/UNIX Windows, consultez la section Téléchargements de Java SE.

Définition de la variable Home de Java

Les outils de ligne de commande dépendant d'une variable d'environnement (`JAVA_HOME`) pour localiser Java Runtime. Cette variable d'environnement doit être définie sur le chemin d'accès complet du répertoire contenant un sous-répertoire nommé `bin` lequel contient l'exécutable `java` (sur Linux et UNIX) ou l'exécutable `java.exe` (sur Windows).

Pour définir la variable Home de Java

1. Définissez la variable Home de Java.
   - Sur Linux et UNIX, entrez la commande suivante :

     ```
     $ export JAVA_HOME=<PATH>
     ```

   - Sur Windows, entrez la commande suivante:

     ```
     C:\> set JAVA_HOME=<PATH>
     ```

2. Confirmez le paramètre de chemin en exécutant **`$JAVA_HOME/bin/java -version`** et en vérifiant la sortie.
   - Sur Linux/UNIX, vous verrez des résultats similaires à ce qui suit :

     ```
     $ $JAVA_HOME/bin/java -version
     java version "1.6.0_23"
     Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
     Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
     ```

   - Sur Windows, vous verrez des résultats similaires à ce qui suit :

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

Téléchargement des outils de ligne de commande

Les outils de ligne de commande sont disponibles sous la forme d'un fichier ZIP, sur le site web ElastiCache Developer Tools. Ces outils sont écrits en Java et incluent des scripts shell pour Windows 2000/ XP/Vista/Windows 7, Linux/UNIX et Mac OSX. Le fichier ZIP file est autonome et aucune installation n'est requise ; téléchargez simplement le fichier zip et décompressez-le dans un répertoire sur votre machine locale.

Installation des outils

Les outils de ligne de commande dépendant d'une variable d'environnement (AWS_ELASTICACHE_HOME) pour localiser les bibliothèques prises en charge. Vous devez définir cette variable d'environnement avant d'utiliser ces outils. Définissez-la sur le chemin du répertoire dans lequel vous avez décompressé ces outils de ligne de commande. Ce répertoire est nommé ElastiCacheCli -A.b.nnnn (A, B et n sont des numéros de version/version) et contient des sous-répertoires nommés bin et lib.

Pour définir la variable d'environnement AWS_ELASTICACHE_HOME

- Ouvrez une fenêtre de ligne de commande et entrez l'une des commandes suivantes pour définir la variable d'environnement AWS_ELASTICACHE_HOME.
  - Sur Linux et UNIX, entrez la commande suivante :

    ```
    $ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
    ```

  - Sur Windows, entrez la commande suivante:

    ```
    C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
    ```

Pour simplifier l'utilisation des outils, nous vous recommandons d'ajouter le répertoire BIN des outils à votre variable PATH système. Le reste de ce guide suppose que le répertoire BIN se trouve dans votre chemin système.

Pour ajouter le répertoire BIN des outils à votre chemin système

- Entrez les commandes suivantes pour ajouter le répertoire BIN des outils à votre chemin système PATH.
  - Sur Linux et UNIX, entrez la commande suivante :

    ```
    $ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
    ```

  - Sur Windows, entrez la commande suivante:

    ```
    C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
    ```

> **Note**
>
> Les variables d'environnement Windows sont réinitialisées lorsque vous fermez la fenêtre de commande. Vous pouvez également si vous le souhaitez les configurer définitivement. Pour plus d'informations, consultez la documentation relative à votre version de Windows.

> **Note**
>
> Les chemins qui comportent un espace doivent être indiqués entre guillemets, par exemple : "C:\Program Files\Java"

Indication des informations d'identification pour les outils

Les outils de ligne de commande ont besoin de la clé AWS d'accès et de la clé d'accès secrète fournies avec votre AWS compte. Vous pouvez les obtenir en utilisant la ligne de commande ou à partir du fichier d'informations d'identification situé sur votre système local.

Le déploiement inclut un fichier modèle $ {AWS_ELASTICACHE_HOME}/credential-file-path.template que vous devez modifier avec vos informations. Voici le contenu du fichier modèle :

```
AWS AccessKeyId=<Write your AWS access ID>
AWS SecretKey=<Write your AWS secret key>
```

> **Important**
>
> Sous UNIX, limitez les autorisations au propriétaire du fichier d'identification :
>
> ```
> $ chmod 600 <the file created above>
> ```

Lors de la configuration du fichier d'informations d'identification, vous devez définir la variable d'environnement AWS_CREDENTIAL_FILE afin que les ElastiCache outils puissent trouver vos informations.

Pour définir la variable d'environnement AWS_CREDENTIAL_FILE

1. Définissez la variable d'environnement :
   - Sur Linux et UNIX, mettez à jour la variable à l'aide de la commande suivante :

     ```
     $ export &AWS;_CREDENTIAL_FILE=<the file created above>
     ```

   - Sous Windows, définissez la variable à l'aide de la commande suivante :

     ```
     C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
     ```

2. Vérifiez que votre configuration fonctionne correctement, exécutez la commande suivante :

   ```
   elasticache --help
   ```

   Vous devriez voir la page d'utilisation pour toutes les commandes ElastiCache.

Variables d'environnement

Les variables d'environnement peuvent être utiles pour le scripting, la configuration de valeurs par défaut ou leur substitution temporaire.

Outre la variable d'environnement AWS_CREDENTIAL_FILE, la plupart des outils d'API inclus dans l'interface de ligne de ElastiCache commande prennent en charge les variables suivantes :

- EC2_REGION — AWS Région à utiliser.
- AWS_ELASTICACHE_URL — URL à utiliser pour l'appel du service. Il n'est pas nécessaire de spécifier un point de terminaison régional différent si EC2 _REGION est spécifié ou si le paramètre --region est transmis.

Les exemples suivants montrent comment définir la variable environnementale EC2 _REGION pour configurer la région utilisée par les outils d'API :

Linux, OS X ou Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Messages ElastiCache d'erreur Amazon

Les messages d'erreur suivants sont renvoyés par Amazon ElastiCache. Vous pouvez recevoir d'autres messages d'erreur renvoyés par ElastiCache d'autres AWS services ou par Valkey, Memcached ou Redis OSS. Pour obtenir une description des messages d'erreur provenant de sources autres que ElastiCache, consultez la documentation de la source qui génère le message d'erreur.

- Cluster node quota exceeded
- Customer's node quota exceeded
- Manual snapshot quota exceeded
- Insufficient cache cluster capacity

Message d'erreur : quota de nœud de cluster dépassé. Chaque cluster peut avoir un maximum de %n nœuds dans cette région.

> Cause : vous avez essayé de créer ou de modifier un cluster de sorte que le cluster ait plus de %n nœuds.

Solution : modifiez votre demande afin que le cluster n'ait pas plus de %n nœuds. Ou, si vous avez besoin de plus de %n nœuds, faites votre demande à l'aide du formulaire de demande Amazon ElastiCache Node.

Pour plus d'informations, consultez Amazon ElastiCache Limits dans Référence générale d'Amazon Web Services.

Messages d'erreur : quota de nœud client dépassé. Vous pouvez avoir un maximum de %n nœuds dans cette région Ou, Vous avez déjà atteint votre quota de %s nœuds dans cette région.

Cause : vous avez essayé de créer ou de modifier un cluster de sorte que votre compte ait plus de %n nœuds sur tous les clusters de cette région.

Solution : modifiez votre demande pour que tous les nœuds de la région sur tous les clusters pour ce compte ne dépassent pas %n. Ou, si vous avez besoin de plus de %n nœuds, faites votre demande à l'aide du formulaire de demande Amazon ElastiCache Node.

Pour plus d'informations, consultez Amazon ElastiCache Limits dans Référence générale d'Amazon Web Services.

Messages d'erreur : le nombre maximal d'instantanés manuels pour ce cluster pris en l'espace de 24 heures a été atteint ou le nombre maximal d'instantanés manuels pour ce nœud pris en l'espace de 24 heures a été atteint son quota de %n

Cause : vous avez essayé de prendre un instantané manuel d'un cluster alors que vous aviez déjà pris le nombre maximal d'instantanés manuels autorisés sur une période de 24 heures.

Solution : attendre 24 heures pour faire un autre instantané manuel du cluster. Ou, si vous avez besoin de prendre un instantané manuel dès maintenant, prenez-en un d'un autre cluster ayant les mêmes données, par exemple un nœud différent dans un cluster.

Messages d'erreur : InsufficientCacheClusterCapacity

Cause : actuellement, AWS ne dispose pas de capacité à la demande suffisante pour répondre à votre demande.

Solution :

- Attendez quelques minutes, puis renvoyez votre demande. La capacité peut changer fréquemment.
- Envoyez une nouvelle demande avec un nombre inférieur de nœuds ou de partitions (groupes de nœuds). Par exemple, si vous avez fait une seule demande pour lancer 15 nœuds, essayez de faire 3 demandes pour lancer 5 nœuds ou 15 demandes pour lancer 1 nœud à la place.
- Si vous lancez un cluster, envoyez une nouvelle demande sans spécifier de zone de disponibilité.
- Si vous lancez un cluster, envoyez une nouvelle demande en utilisant un type de nœud différent (que vous pourrez augmenter à un stade ultérieur). Pour de plus amples informations, veuillez consulter Dimensionnement ElastiCache.

Notifications

Cette rubrique couvre ElastiCache les notifications susceptibles de vous intéresser. Une notification est une situation ou un événement qui, dans la plupart des cas, est temporaire et dure uniquement jusqu'à ce qu'une solution soit trouvée et mise en œuvre. Les notifications ont généralement une date de début et une date de résolution, après lesquelles la notification n'est plus pertinente. Une notification ne vous concerne pas nécessairement. Nous recommandons une directive d'implémentation qui, si elle est suivie, permet d'améliorer les performances de votre cluster.

Les notifications n'annoncent pas de ElastiCache fonctionnalités nouvelles ou améliorées.

ElastiCache Notifications générales

Il n'y a actuellement aucune ElastiCache notification en suspens qui ne soit pas spécifique au moteur.

ElastiCache pour les notifications Memcached

Les ElastiCache notifications suivantes sont spécifiques au moteur Memcached.

ElastiCache pour les notifications spécifiques à Memcached

- Alerte : crawler LRU Memcached entraîne des problèmes de segmentation

Alerte : crawler LRU Memcached entraîne des problèmes de segmentation

Date d'alerte : 28 février 2017

Dans certaines circonstances, votre cluster peut être instable en raison d'un problème de segmentation dans le crawler LRU Memcached. Il s'agit d'un problème du moteur Memcached connu depuis quelque temps déjà. Le problème est apparu dans Memcached 1.4.33 lorsque le crawler LRU a été activé par défaut.

Si vous rencontrez ce problème, nous vous recommandons de désactiver le crawler LRU jusqu'à ce qu'un correctif soit disponible. Pour cela, utilisez `lru_crawler disable` au niveau de la ligne de commande ou modifiez la valeur de paramètre `lru_crawler` (recommandé).

Date de résolution :

Résolution :

ElastiCache pour les notifications spécifiques à Redis OSS

Aucune notification Redis OSS n'est actuellement en suspens ElastiCache .

ElastiCache Historique de la documentation

- Version de l'API : 2015-02-02
- Dernière date de mise à jour de la documentation : 27 novembre 2023

Le tableau suivant décrit les modifications importantes apportées à chaque version du guide de ElastiCache l'utilisateur après mars 2018. Pour recevoir les notifications sur les mises à jour de cette documentation, vous pouvez vous abonner au Flux RSS.

ElastiCache Mises à jour récentes

Modification	Description	Date
Support de la mise à ElastiCache l'échelle verticale à la demande de Memcached et de la mise à l'échelle horizontale automatisée	ElastiCache avec Memcached prend désormais en charge la mise à l'échelle verticale à la demande et la mise à l'échelle horizontale automatisée. Cela inclut une mise à jour de la politique pour l'action `elasticache:ModifyCacheCluster`	10 avril 2025
Support pour ElastiCache Valkey	ElastiCache supporte désormais Valkey. Valkey 7.2.6 est compatible avec Redis OSS 7.2 Pour plus d'informations, consultez Valkey.	8 octobre 2024
Dimensionnez les nœuds réservés flexibles	ElastiCache prend désormais en charge les nœuds réservés Size flexible. Pour plus d'informations, consultez Amazon ElastiCache Pricing.	1er octobre 2024

Support ElastiCache pour Memcached 1.6.22	ElastiCache pour Memcached supporte désormais Memcached 1.6.22. Il inclut les correctifs de bogue cumulatifs à partir de la version 1.6.18. Pour plus d'informations, consultez Memcached Version 1.6.22.	10 janvier 2024
ElastiCache pour Redis OSS : prise en charge supplémentaire pour des tailles de nœuds C7gn supplémentaires	ElastiCache pour Redis OSS, ajout de la prise en charge de tailles de nœuds C7gn supplémentaires.	10 janvier 2024

ElastiCache pour Redis OSS prend désormais en charge la création de caches sans serveur	Vous pouvez désormais créer des caches sans serveur, ce qui simplifie la gestion des caches et leur mise à l'échelle instantanée pour prendre en charge les applications les plus exigeantes. Pour plus d'informations, consultez Choix entre deux options de déploiement. Dans le cadre de cette fonctionnalité, de nouvelles autorisations ont été ajoutées à `ElastiCacheServiceRolePolicy` et `AmazonElastiCacheFullAccess` pour permettre d'associer des caches sans serveur à des points de terminaison de VPC gérés. En outre, des autorisations ont été ajoutées pour prendre en charge une expérience de console révisée en utilisant la politique `AmazonElastiCacheFullAccess` .	27 novembre 2023

ElastiCache pour Memcached prend désormais en charge la création de caches sans serveur	Vous pouvez désormais créer des caches sans serveur, ce qui simplifie la gestion des caches et leur mise à l'échelle instantanée pour prendre en charge les applications les plus exigeantes. Pour plus d'informations, consultez Choix entre deux options de déploiement. Dans le cadre de cette fonctionnalité, de nouvelles autorisations ont été ajoutées à `ElastiCacheServiceRolePolicy` et `AmazonElastiCacheFullAccess` pour permettre d'associer des caches sans serveur à des points de terminaison de VPC gérés. En outre, des autorisations ont été ajoutées pour prendre en charge une expérience de console révisée en utilisant la politique `AmazonElastiCacheFullAccess` .	27 novembre 2023
ElastiCache pour Redis OSS prend désormais en charge la modification du mode cluster	Vous pouvez désormais migrer des clusters du mode cluster désactivé (CMD) vers le mode cluster activé (CME). Pour plus d'informations, consultez Modifying cluster mode (Modification du mode cluster).	11 mai 2023

ElastiCache pour Redis OSS prend désormais en charge la modification des paramètres de chiffrement en transit	Vous pouvez désormais modifier la configuration TLS de vos clusters Redis OSS sans avoir à reconstruire ou à reprovisionner les clusters ou à affecter la disponibilité des applications. Pour plus d'informations, consultez la section Activation du chiffrement en transit sur un cluster existant.	28 décembre 2022
ElastiCache pour Redis OSS prend désormais en charge l'authentification des utilisateurs à l'aide d'IAM	L'authentification IAM vous permet d'authentifier une connexion à Redis OSS à ElastiCache l'aide AWS d'identités IAM. Cela vous permet de renforcer votre modèle de sécurité et de simplifier de nombreuses tâches administratives de sécurité. Pour plus d'informations, consultez Authenticating with IAM (Authentification avec IAM).	16 novembre 2022
ElastiCache pour Redis OSS prend désormais en charge Redis OSS 7	Cette version apporte plusieurs nouvelles fonctionnalités à Amazon ElastiCache pour Redis OSS : fonctions Redis OSS, améliorations de l'ACL et Sharded Pub/Sub. Pour plus d'informations, consultez ElastiCache Redis OSS version 7.0.	8 novembre 2022

ElastiCache pour Redis OSS prend désormais en charge IPV6	ElastiCache prend en charge les versions 4 et 6 du protocole Internet (IPv4 et IPv6), ce qui vous permet de configurer votre cluster pour qu'il accepte uniquement IPv4 les connexions, uniquemen t IPv6 les connexions ou les deux IPv4 et les IPv6 connexions (double pile). IPv6 est pris en charge pour les charges de travail utilisant le moteur Redis OSS version 6.2 et ultérieure sur toutes les instances créées sur le système Nitro. Il n'y a pas de frais supplémentaires pour y ElastiCache accéder IPv6. Pour plus d'informa tions, consultez Choosing a network type (Choisir un type de réseau).	7 novembre 2022

ElastiCache pour Memcached supporte désormais IPV6	ElastiCache prend en charge les versions 4 et 6 du protocole Internet (IPv4 et IPv6), ce qui vous permet de configurer votre cluster pour qu'il accepte uniquement IPv4 les connexions, uniquemen t IPv6 les connexions ou les deux IPv4 et les IPv6 connexions (double pile). IPv6 est pris en charge pour les charges de travail utilisant le moteur Memcached version 1.6.6 et ultérieure sur toutes les instances créées sur le système Nitro. Il n'y a pas de frais supplémentaires pour y ElastiCache accéder IPv6. Pour plus d'informa tions, consultez Choosing a network type (Choisir un type de réseau).	7 novembre 2022
ElastiCache pour Memcached prend désormais en charge le chiffrement en transit	Le chiffrement en transit est une fonction facultative qui vous permet d'accroître la sécurité de vos données aux endroits les plus vulnérables lorsqu'elles sont en transit d'un endroit à un autre. Il est pris en charge sur les versions Memcached 1.6.12 et ultérieur es. Pour plus d'informations, consultez la section Chiffreme nt ElastiCache en transit (TLS).	26 mai 2022

ElastiCache pour Redis OSS prend désormais en charge le format natif de notation JavaScript d'objet (JSON)	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 désériali ser. Pour plus d'informations, consultez Mise en route avec JSON.	25 mai 2022

ElastiCache prend désormais en charge PrivateLink	AWS PrivateLink vous permet d'accéder aux opération s d' ElastiCache API de manière privée sans passerell e Internet, appareil NAT, connexion VPN ou connexion AWS Direct Connect. Pour plus d'informations, consultez les points de terminaison Amazon ElastiCache API et interface VPC (AWS PrivateLi nk) pour Redis OSS ou Amazon API ElastiCache et points de terminaison VPC d'interface () pour Memcached .AWS PrivateLink	24 janvier 2022

ElastiCache pour Redis OSS prend désormais en charge Redis OSS 6.2 et Data Tiering	Amazon ElastiCache pour Redis OSS présente la prochaine version du moteur Redis OSS pris en charge par Amazon. ElastiCache ElastiCache pour Redis OSS 6.2 inclut des améliorations de performances pour les clusters compatibles TLS utilisant des types de nœuds x86 de 8 V CPUs ou plus ou des types de nœuds Graviton2 de 4 V ou plus. CPUs ElastiCac he pour Redis OSS introduit également la hiérarchisation des données. 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'informa tions, consultez ElastiCac he Redis OSS version 6.2 (améliorée) et Data tiering.	23 novembre 2021

Prise en charge d'Auto Scaling	ElastiCache pour Redis OSS supporte désormais Auto Scaling. ElastiCache pour Redis OSS, le dimension nement automatique est la possibilité d'augmenter ou de diminuer automatiquement les partitions ou les répliques souhaités dans votre service ElastiCache pour Redis OSS. ElastiCache utilise le service Application Auto Scaling pour fournir cette fonctionn alité. Pour plus d'informa tions, consultez Auto Scaling ElastiCache pour les clusters Redis OSS.	19 août 2021
Support pour la livraison des journaux de lenteur de Redis OSS	ElastiCache vous permet désormais de diffuser Redis OSS SLOWLOG vers l'une des deux destinati ons suivantes : Amazon Data Firehose ou Amazon Logs. CloudWatch Pour de plus amples informations, veuillez consulter Livraison de journaux.	22 avril 2021

Prise en charge de l'étiquetage des ressources et des clés de condition	ElastiCache prend désormais en charge le balisage pour vous aider à gérer vos clusters et autres ElastiCache ressources. Pour plus d'informations, consultez la section Marquage de vos ElastiCache ressources. ElastiCache introduit également la prise en charge des clés de condition. Vous pouvez spécifier des conditions pour déterminer comment une stratégie IAM prend effet. Pour de plus amples informations, veuillez consulter Utilisation de ces clés de condition.	7 avril 2021
Prise en charge de l'étiquetage des ressources et des clés de condition	ElastiCache prend désormais en charge le balisage pour vous aider à gérer vos clusters et autres ElastiCache ressources. Pour plus d'informations, consultez la section Marquage de vos ElastiCache ressources. ElastiCache introduit également la prise en charge des clés de condition. Vous pouvez spécifier des conditions pour déterminer comment une stratégie IAM prend effet. Pour de plus amples informations, veuillez consulter Utilisation de ces clés de condition.	7 avril 2021

ElastiCache est désormais disponible sur AWS Outposts	AWS Outposts apporte des AWS services, une infrastructure et des modèles d'exploitation natifs à pratiquement tous les centres de données, espaces de colocation ou installations sur site. Vous pouvez effectuer un déploiement ElastiCache sur Outposts pour configurer, exploiter et utiliser le cache sur site, comme vous le feriez dans le cloud. Pour plus d'informations, consultez la section Utilisation des Outposts.	8 octobre 2020
ElastiCache prend désormais en charge Redis OSS 6	Amazon ElastiCache pour Redis OSS présente la prochaine version du moteur Redis OSS pris en charge par Amazon. ElastiCache Cette version inclut l'authentification des utilisateurs avec le contrôle d'accès basé sur les rôles, la prise en charge sans version, la mise en cache côté client et des améliorations opérationnelles significatives. Pour plus d'informations, consultez ElastiCache Redis OSS version 6.0 (améliorée).	7 octobre 2020

ElastiCache prend désormais en charge les Zones Locales	Une zone locale est une extension d'une AWS région géographiquement proche de vos utilisateurs. Vous pouvez étendre n'importe quel cloud privé virtuel (VPC) d'une AWS région parent à des zones locales en créant un nouveau sous-réseau et en l'affectant à une zone locale. Pour de plus amples informations, veuillez consulter Utilisation de Local Zones.	25 septembre 2020
ElastiCache pour Redis OSS prend désormais en charge le dimensionnement de votre environnement Redis OSS Cluster jusqu'à 500 nœuds ou 500 partitions	Le mode Redis OSS Cluster permet des configurations que vous pouvez utiliser pour partitionner vos données sur plusieurs partitions et offre une évolutivité, des performances et une disponibilité améliorées. Cette fonctionnalité est disponible sur Amazon ElastiCache pour les versions 5.0.6 et ultérieures de Redis OSS dans toutes les AWS régions et pour tous les environnements de cluster Redis OSS existants et nouveaux ElastiCache . Pour plus d'informations, consultez la section Nœuds et partitions Redis OSS.	13 août 2020

ElastiCache prend désormais en charge les autorisations au niveau des ressources	Vous pouvez désormais restreindre l'étendue des autorisations d'un utilisateur en spécifiant ElastiCache des ressources dans une politique AWS Identity and Access Management (IAM). Pour plus d'informations, veuillez consulter Autorisations au niveau des ressources.	12 août 2020
ElastiCache pour Redis OSS ajoute des métriques Amazon CloudWatch supplémentaires	ElastiCache pour Redis OSS prend désormais en charge de nouvelles CloudWatch métriques, notamment `PubSubCmds` `etHyperLogLogBasedCmds`. Pour une liste complète, voir Metrics for Redis OSS.	10 juin 2020
ElastiCache prend désormais en charge la mise à jour automatique des clusters ElastiCache	Amazon prend ElastiCache désormais en charge la mise à jour automatique des ElastiCache clusters une fois la « date limite d'application recommandée » passée pour la mise à jour du service. ElastiCache utilisera votre fenêtre de maintenance pour planifier la mise à jour automatique des clusters applicables. Pour plus d'informations, consultez Self-service Updates (Mises à jour en libre-service).	13 mai 2020

ElastiCache pour Redis OSS prend désormais en charge le Global Datastore pour Redis OSS	La fonctionnalité Global Datastore for Redis OSS permet une réplication entièrement gérée, rapide, fiable et sécurisée entre les régions AWS . Grâce à cette fonctionnalité, vous pouvez créer des clusters de répliques de lecture entre régions ElastiCache pour Redis OSS afin de permettre des lectures à faible latence et une reprise après sinistre dans toutes les régions. AWS Vous pouvez créer, modifier et décrire un magasin de données global. Vous pouvez également ajouter ou supprimer des AWS régions de votre banque de données mondiale et promouvoir une AWS région en tant que principale au sein d'une banque de données mondiale. Pour plus d'informations, voir Réplication entre AWS régions à l'aide d'une banque de données globale.	16 mars 2020
ElastiCache pour Redis OSS prend désormais en charge la version 5.0.6 de Redis OSS	Pour plus d'informations, consultez ElastiCache Redis OSS version 5.0.6 (améliorée).	18 décembre 2019

Amazon prend ElastiCache désormais en charge les nœuds de cache standard T3	Vous pouvez désormais lancer la nouvelle génération de nœuds de cache standard T3 à usage général sur Amazon. ElastiCache Les instances EC2 T3 Standard d'Amazon fournissent un niveau de référence en matière de performances du processeur, avec la possibilité d'augmenter l'utilisation du processeur à tout moment jusqu'à épuisement des crédits accumulés. Pour plus d'informations, consultez Types de nœuds pris en charge.	12 novembre 2019

Amazon prend ElastiCache désormais en charge la modification du jeton AUTH sur un serveur Redis OSS existant ElastiCache	ElastiCache pour Redis OSS 5.0.6 vous permet désormais de modifier les jetons d'authentification en définissant et en faisant pivoter de nouveaux jetons. Vous pouvez désormais modifier des jetons actifs pendant qu'ils sont utilisés. Vous pouvez également ajouter de tout nouveaux jetons aux clusters existants activés avec le chiffrement en transit qui étaient auparavant configurés sans jetons d'authentification. Il s'agit d'un processus en deux étapes par lequel vous pouvez définir et renouveler le jeton sans interrompre les demandes du client. Cette fonctionnalité n'est actuellement pas prise en charge sur AWS CloudFormation. Pour plus d'informations, consultez Authentification des utilisateurs à l'aide de la commande Redis OSS AUTH.	30 octobre 2019

Amazon prend ElastiCache désormais en charge la migration de données en ligne depuis Redis OSS sur Amazon EC2	Vous pouvez désormais utiliser la migration en ligne pour migrer vos données depuis Redis OSS auto-hébergé sur Amazon vers EC2 Amazon. ElastiCache Pour plus d'informations, consultez la section Migration en ligne vers ElastiCache.	28 octobre 2019
ElastiCache pour Redis OSS introduit la mise à l'échelle verticale en ligne pour le mode Redis OSS Cluster.	Vous pouvez désormais augmenter ou réduire votre cluster Redis OSS fragmenté à la demande. ElastiCache for Redis OSS redimensionne votre cluster en modifiant le type de nœud, tandis que le cluster continue de rester en ligne et de répondre aux demandes entrantes. Pour plus d'informations, veuillez consulter Dimensionnement vertical en ligne en modifiant le type de nœud.	20 août 2019

ElastiCache pour Redis OSS permet désormais aux utilisateurs d'utiliser un seul point de terminaison de lecteur pour votre cluster Amazon ElastiCache for Redis OSS.	Cette fonctionnalité vous permet de diriger tout le trafic de lecture vers votre ElastiCache cluster Redis OSS via un point de terminaison unique au niveau du cluster afin de tirer parti de l'équilibrage de charge et d'une disponibilité accrue. Pour plus d'informations, veuillez consulter Recherche de points de terminaison de connexion.	13 juin 2019
ElastiCache pour Redis OSS permet désormais aux utilisateurs d'appliquer les mises à jour de service selon leur propre calendrier	Cette fonctionnalité vous permet de choisir d'appliquer les mises à jour du service disponibles lorsque vous le souhaitez et pas seulement pendant les fenêtres de maintenance. Cela permettra de minimiser les interruptions de service, en particulier pendant les pics d'activité, et de garantir votre conformité si votre cluster est soumis à des programmes de conformité ElastiCache pris en charge. Pour plus d'informations, consultez les sections Mises à jour en libre-service sur Amazon ElastiCache et Validation de conformité pour Amazon ElastiCache.	4 juin 2019

ElastiCache Offres d'instances réservées standard : initiale partielle, mise initiale totale et absence d'instance initiale.	Les instances réservées vous permettent de réserver une ElastiCache instance Amazon pour une durée d'un ou trois ans en fonction du type d'instance et de AWS la région. Pour plus d'informations, consultez Gestion des coûts avec les nœuds réservés.	18 janvier 2019
ElastiCache pour la prise en charge de Redis OSS pour un maximum de 250 nœuds par cluster Redis OSS	La limite de nœuds ou de partitions peut être augmentée jusqu'à un maximum de 250 par cluster ElastiCache Redis OSS. Pour plus d'informations, consultez Partitions.	19 novembre 2018
ElastiCache pour la prise en charge de Redis OSS pour le basculement automatique, la sauvegarde et la restauration sur tous les nœuds T2	ElastiCache pour Redis OSS introduit la prise en charge du basculement automatique, de la création de snapshots, ainsi que de la sauvegarde et de la restauration sur tous les nœuds T2. Pour plus d'informations, consultez ElastiCache Redis OSS Backup and Restore and Snapshot.	19 novembre 2018

ElastiCache pour le support de Redis OSS pour les nœuds M5 et R5	ElastiCache pour Redis OSS prend désormais en charge les nœuds M5 et R5, les types d'instances à usage général et optimisés pour la mémoire basés sur le système Nitro. AWS Pour plus d'informations, consultez Types de nœuds pris en charge.	23 octobre 2018
Prise en charge de la modification dynamique du nombre de réplicas en lecture	ElastiCache pour Redis OSS a ajouté la prise en charge de l'ajout et de la suppression de répliques de lecture dans n'importe quel cluster, sans interruption de service du cluster. Pour plus d'informations sur ces modifications et sur les autres modifications apportées à cette version, consultez la section Modification du nombre de répliques dans le guide de l'utilisateur ElastiCache pour Redis OSS. Voir également DecreaseReplicaCountet IncreaseReplicaCountdans la référence de ElastiCache l'API.	17 septembre 2018
Certification de conformité au programme FedRAMP	ElastiCache pour Redis OSS est désormais certifié conforme à FedRAMP. Pour plus d'informations, consultez la section Validation de conformité pour Amazon ElastiCache.	30 août 2018

Mises à niveau du moteur Valkey ou Redis OSS (mode cluster activé)	Amazon ElastiCache pour Redis OSS a ajouté la prise en charge de la mise à niveau des versions du moteur Valkey ou Redis OSS (mode cluster activé). Pour plus d'informations, consultez Mise à niveau des versions de moteur.	20 août 2018
Certifications de conformité à PCI DSS	ElastiCache pour Redis OSS est désormais certifié conforme à la norme PCI DSS. Pour plus d'informations, consultez la section Validation de conformité pour Amazon ElastiCache.	5 juillet 2018
Support ElastiCache pour Redis OSS 4.0.10	ElastiCache pour Redis OSS prend désormais en charge Redis OSS 4.0.10, y compris le chiffrement et le redimensionnement du cluster en ligne dans une seule version. Pour plus d'informations, consultez ElastiCache Redis OSS version 4.0.10 (améliorée).	14 juin 2018

Nouvelle structure du guide de l'utilisateur	Le guide de ElastiCac he l'utilisateur unique est maintenant restructuré de sorte qu'il existe des guides d'utilisation distincts pour Redis OSS (pour le guide de l'utilisateur de Redis OSS) et ElastiCache pour Memcached (ElastiCache pour le guide de l'utilisateur Memcached). La structure de documentation de la section AWS CLI Comma nd Reference : elasticache et celle de l'Amazon ElastiCac he API Reference restent inchangées.	20 avril 2018
Support pour Engine CPUUtiliz ation Metric	ElastiCache pour Redis OSS a ajouté une nouvelle métrique`EngineCPU Utilization`, qui indique le pourcentage de capacité de votre processeur actuellem ent utilisé. Pour plus d'informa tions, consultez Metrics for Redis OSS.	9 avril 2018

Le tableau suivant décrit les modifications importantes apportées au guide de l'ElastiCache utilisateur avant mars 2018.

Modification	Description	Date de modification
Prise en charge de la région Asie-Pacifique (Osaka-Local).	ElastiCache ajout d'un support pour la région Asie-Pacifique (local d'Osaka). La région Asie-Pacifique (Osaka) prend actuellement en charge une seule zone de disponibilité et est accessible sur invitatio	12 février 2018

Modification	Description	Date de modification
	n uniquement. Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	
Prise en charge de l'UE (Paris).	ElastiCache support supplémentaire pour la région UE (Paris). Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	18 décembre 2017
Prise en charge de la région Chine (Ningxia)	Amazon ElastiCache a ajouté le support pour la région Chine (Ningxia). Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	11 décembre 2017
Prise en charge des rôles liés aux services	Cette version ElastiCache ajoute la prise en charge des rôles liés au service (SLR). Pour plus d'informations, consultez les ressources suivantes : • Utilisation de rôles liés à un service pour Amazon ElastiCache • Configurez vos autorisations (nouveaux ElastiCache utilisateurs uniquement)	7 décembre 2017

Modification	Description	Date de modification
Prise en charge des types de nœud R4	Cette version prend en charge les types de nœuds R4 ElastiCache supplémentaires dans toutes les AWS régions prises en charge par ElastiCache. Vous pouvez acheter des types de nœud R4 à la demande ou des nœuds de cache réservés. Pour plus d'informations, consultez les ressources suivantes : • Types de nœuds de cache pris en charge • Paramètres propres au type de nœud Memcached • Paramètres spécifiques au type de nœud Redis OSS	le 20 novembre 2017
ElastiCache pour Redis OSS 3.2.10 et prise en charge du repartage en ligne	Amazon ElastiCache pour Redis OSS ajoute la prise en charge de ElastiCache Redis OSS 3.2.10. ElastiCache pour Redis OSS introduit également le redimensionnement du cluster en ligne pour ajouter ou supprimer des fragments du cluster tout en continuant à traiter les demandes entrantes. I/O Pour plus d'informations, consultez les ressources suivantes : • Redimensionnement de cluster en ligne • Repartage en ligne pour Valkey ou Redis OSS (mode cluster activé)	9 novembre 2017

Modification	Description	Date de modification
Éligibilité HIPAA	ElastiCache pour Redis OSS, la version 3.2.6 est désormais certifiée conforme à l'HIPAA lorsque le chiffrement est activé sur votre cluster. Pour plus d'informations, consultez les ressources suivantes : • Validation de conformité pour Amazon ElastiCache • Sécurité des données sur Amazon ElastiCache	2 novembre 2017
ElastiCache pour Redis OSS 3.2.6 et prise en charge du chiffrement	ElastiCache ajoute le support ElastiCache pour Redis OSS 3.2.6, qui inclut deux fonctionnalités de chiffrement : • Le chiffrement en transit chiffre vos données lorsqu'elles sont transférées, par exemple entre les nœuds d'un cluster ou entre un cluster et votre application. • Le chiffrement au repos chiffre vos données sur disque au cours de la synchronisation et des opérations de sauvegarde. Pour plus d'informations, consultez les ressources suivantes : • Sécurité des données sur Amazon ElastiCache • Moteurs et versions pris en charge	25 octobre 2017

Modification	Description	Date de modification
Rubrique Modèles de connexion	ElastiCache la documentation ajoute une rubrique couvrant les différents modèles d'accès à un ElastiCache cluster dans un Amazon VPC. Pour plus d'informations, consultez Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC dans le Guide de l'utilisateur ElastiCache .	24 avril 2017
Prise en charge de Memcached 1.4.34	ElastiCache ajoute le support de Memcached version 1.4.34, qui intègre un certain nombre de correctifs aux versions antérieures de Memcached. Pour plus d'informations, consultez les notes de mise à jour de Memcached 1.4.34 sur Memcached on. GitHub	10 avril 2017
Support de test du basculement automatique	ElastiCache ajoute la prise en charge du test du basculement automatique sur les clusters Redis OSS qui prennent en charge la réplication. Pour plus d'informations, consultez les ressources suivantes : • Test du basculement automatique dans le guide de l'utilisateur ElastiCache . • TestFailover dans la Référence d'API ElastiCac he • test-failover dans le manuel Référence de l'AWS CLI .	4 avril 2017

Modification	Description	Date de modification
Restauration améliorée de Redis OSS	ElastiCache ajoute une sauvegarde et une restauration Redis OSS améliorées avec le redimensionnement du cluster. Cette fonctionn alité permet la restauration d'une sauvegarde sur un cluster avec un nombre de partitions différent de celui du cluster utilisé pour créer la sauvegard e. (Pour l'API et la CLI, cette fonctionnalité peut restaurer un nombre différent de groupes de nœuds au lieu d'un nombre différent de partition s.) Cette mise à jour prend également en charge différentes configurations de slot Redis OSS. Pour de plus amples informations, veuillez consulter Restauration à partir d'une sauvegarde dans un nouveau cache.	15 mars 2017
Nouveau paramètre de gestion de la mémoire Redis OSS	ElastiCache ajoute un nouveau paramètre Redis OSS`reserved-memory-percent`, qui facilite la gestion de votre mémoire réservée. Ce paramètre est disponible sur toutes les versions de ElastiCache pour Redis OSS. Pour plus d'informa tions, consultez les ressources suivantes : • Gestion de la mémoire réservée pour Valkey et Redis OSS • Nouveaux paramètres pour Redis OSS 3.2.4	15 mars 2017
Prise en charge de Memcached 1.4.33	ElastiCache ajoute le support pour la version 1.4.33 de Memcached. Pour plus d'informations, consultez les ressources suivantes : • ElastiCache version 1.4.33 pour Memcached • Paramètres ajoutés Memcached 1.4.33	20 décembre 2016

Modification	Description	Date de modification
Prise en charge pour la région UE Ouest (Londres)	ElastiCache ajoute le soutien à la région UE (Londres). Seuls les types de nœud T2 et M4 sont actuellement pris en charge. Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	13 décembre 2016
Prise en charge pour la région Canada (Montréal)	ElastiCache ajoute un soutien à la région du Canada (Montréal). Seuls les nœuds de type M4 et T2 sont actuellement pris en charge dans cette AWS région. Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	8 décembre 2016
Prise en charge des types de nœud M4 et R3	ElastiCache ajoute la prise en charge des types de nœuds R3 et M4 dans la région d'Amérique du Sud (São Paulo) et des types de nœuds M4 dans la région de Chine (Pékin). Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	1er novembre 2016
Prise en charge de la région USA Est 2 (Ohio)	ElastiCache ajoute le support pour la région USA Est (Ohio) (us-east-2) avec les types de nœuds M4, T2 et R3. Pour plus d'informations, consultez les ressources suivantes : • Régions prises en charge • Types de nœuds de cache pris en charge	17 octobre 2016

Modification	Description	Date de modification
Support pour Redis OSS Cluster	ElastiCache ajoute le support pour Redis OSS Cluster (amélioré). Les clients utilisant Redis OSS Cluster peuvent partitionner leurs données sur un maximum de 15 partitions (groupes de nœuds). Chaque partition prend en charge la réplication, avec jusqu'à 5 réplicas en lecture par partition. Les temps de basculement automatique du cluster Redis OSS sont environ un quart inférieurs à ceux des versions précédentes. Cette version inclut une console de gestion refondue qui utilise la terminologie conforme à celle du secteur. Pour plus d'informations, consultez les ressources suivantes : • Comparaison entre Memcached et Redis OSS • ElastiCache composants et fonctionnalités — notez les sections sur les nœuds, les partitions, les clusters et la réplication. • ElastiCache terminologie	12 octobre 2016

Modification	Description	Date de modification
Prise en charge du type de nœud M4	ElastiCache ajoute la prise en charge de la famille de types de nœuds M4 dans la plupart des AWS régions prises en charge parElastiCache. Vous pouvez acheter des types de nœud M4 à la demande ou des nœuds de cache réservés. Pour plus d'informations, consultez les ressources suivantes : • Types de nœuds de cache pris en charge • Paramètres propres au type de nœud Memcached • Paramètres spécifiques au type de nœud Redis OSS	3 août 2016
Prise en charge de la région de Mumbai	ElastiCache ajoute le support à la région Asie-Pacifique (Mumbai). Pour plus d'informations, consultez les ressources suivantes : • Types de nœuds de cache pris en charge • Paramètres propres au type de nœud Memcached • Paramètres spécifiques au type de nœud Redis OSS	27 juin 2016
Exportation d'instantané	ElastiCache ajoute la possibilité d'exporter un instantané Redis OSS afin que vous puissiez y accéder de l'extérieurElastiCache. Pour plus d'informations, consultez les ressources suivantes : • Exportation d'une sauvegardedans le guide de ElastiCache l'utilisateur Amazon • CopySnapshotdans le Amazon ElastiCache API Reference	26 mai 2016

Modification	Description	Date de modification
Augmentation d'un type de nœud	ElastiCache ajoute la possibilité d'augmenter le type de nœud Redis OSS. Pour de plus amples informations, veuillez consulter Dimensionnement ElastiCache.	24 mars 2016
Mise à niveau simple de moteur	ElastiCache ajoute la possibilité de mettre à niveau facilement votre moteur de cache Redis OSS. Pour de plus amples informations, veuillez consulter Gestion des versions pour ElastiCache.	22 mars 2016
Prise en charge des types de nœud R3	ElastiCache ajoute la prise en charge des types de nœuds R3 dans les régions de Chine (Pékin) et d'Amérique du Sud (São Paulo). Pour plus d'informations, consultez Types de nœuds de cache pris en charge.	16 mars 2016
Accès à ElastiCache l'aide d'une fonction Lambda	Ajout d'un didacticiel sur la configuration d'une fonction Lambda pour y accéder ElastiCache dans un Amazon VPC. Pour de plus amples informati ons, veuillez consulter Autres ElastiCache tutoriels et vidéos.	12 février 2016
Support pour Redis OSS 2.8.24	ElastiCache ajoute le support pour Redis OSS version 2.8.24 avec des améliorations ajoutées depuis Redis OSS 2.8.23. Les améliorations comprennent des correctifs et la prise en charge de la journalisation des adresses de mémoire incorrectes. Pour plus d'informations, consultez les ressources suivantes : • ElastiCache version 2.8.24 pour Redis OSS (améliorée) • Notes de mise à jour de Redis OSS 2.8	20 janvier 2016

Modification	Description	Date de modification
Prise en charge de la région Asie-Pacifique (Séoul)	ElastiCache ajoute le support pour la région Asie-Pacifique (Séoul) (ap-northeast-2) avec les types de nœuds t2, m3 et r3.	6 janvier 2016
Modification de ElastiCache la console Amazon.	Les nouvelles versions de Redis OSS offrant une expérience utilisateur meilleure et plus stable, les versions 2.6.13, 2.8.6 et 2.8.19 de Redis OSS ne sont plus répertoriées dans la console de gestion. ElastiCache Pour les autres options et plus d'informations, consultez Moteurs et versions pris en charge.	15 décembre 2015
Support pour Redis OSS 2.8.23.	ElastiCache ajoute le support pour Redis OSS version 2.8.23 avec des améliorations ajoutées depuis Redis OSS 2.8.22. Les améliorations incluent des correctifs et la prise en charge du nouveau paramètre `close-on-slave-write` qui, s'il est activé, déconnecte les clients qui tentent d'écrire sur un réplica en lecture seule. Pour de plus amples informations, veuillez consulter ElastiCache version 2.8.23 pour Redis OSS (améliorée).	13 novembre 2015

Modification	Description	Date de modification
Support pour Redis OSS 2.8.22.	ElastiCache ajoute le support pour Redis OSS version 2.8.22 avec des améliorations et améliorat ions ElastiCache supplémentaires depuis la version 2.8.21. Ces améliorations sont les suivantes : • Mise en place d'un processus d'enregistrement sans fonction fork qui permet d'enregistrer même lorsque la mémoire disponible est faible, alors que cela aura échoué avec un processus de enregistrement utilisant la fonction fork. • CloudWatch Indicateurs supplémentaires — SaveInProgresset ReplicationBytes. • Pour activer les synchronisations partielles, le paramètre Redis OSS s'applique `repl-backlog-size` désormais à tous les clusters. Pour une liste complète des modifications et plus d'informations, consultez ElastiCache version 2.8.22 pour Redis OSS (améliorée). Cette version de documentation inclut une réorganisation de la documentation et la suppressi on de la documentation de l'interface de ligne de ElastiCache commande (CLI). Pour une utilisation en ligne de commande, reportez-vous à la ligne de AWS commande pour ElastiCache.	28 septembre 2015

Modification	Description	Date de modification
Prise en charge de Memcached 1.4.28	ElastiCache ajoute le support pour la version 1.4.24 de Memcached et les améliorations apportées à Memcached depuis la version 1.4.14. Cette version ajoute la prise en charge de la gestion de cache de type LRU (moins récemment utilisé) en tâche d'arrière-plan, le choix des algorithmes jenkins ou murmur3 en tant qu'algori thme de hachage, de nouvelles commandes et divers correctifs. Pour de plus amples informati ons, veuillez consulter Notes de mise à jour de Memcached. .	27 août 2015
Prise en charge de découvert e automatique Memcached à l'aide de PHP 5.6	Cette version d'Amazon ElastiCache ajoute la prise en charge du client Memcached Auto Discovery pour PHP version 5.6. Pour de plus amples informations, veuillez consulter Compiler le code source du client de ElastiCache cluster pour PHP.	29 juillet 2015
Support pour Redis OSS 2.8.21	ElastiCache ajoute la prise en charge de la version 2.8.21 de Redis OSS et des améliorations apportées à Redis OSS depuis la version 2.8.19. Cette version de Redis OSS inclut plusieurs corrections de bogues. Pour plus d'informations, consultez les notes de mise à jour de Redis OSS 2.8.	29 juillet 2015
Nouveau sujet : Accès ElastiCac he depuis l'extérieur AWS	Ajout d'un nouveau sujet sur la façon d'accéder aux ElastiCache ressources depuis l'extérieur AWS. Pour de plus amples informations, veuillez consulter Accès aux ElastiCache ressources depuis l'extérieur AWS.	9 juillet 2015

Modification	Description	Date de modification
Messages de remplacement de nœud ajoutés	ElastiCache ajoute trois messages relatifs au remplacement planifié des nœudsNodeReplacementScheduled, ElastiCache ElastiCache :NodeReplacementRescheduled, : et ElastiCache :NodeReplacementCanceled. Pour plus d'informations et pour connaître les mesures que vous pouvez prendre lorsqu'il est prévu de remplacer un nœud, consultez ElastiCache la sectionNotifications d'événements Amazon SNS.	11 juin 2015

Modification	Description	Date de modification
Support pour Redis OSS version 2.8.19.	ElastiCache ajoute la prise en charge de la version 2.8.19 de Redis OSS et des améliorations apportées à Redis OSS depuis la version 2.8.6. Les éléments suivants sont pris en charge : • La structure de HyperLogLog données, avec les commandes Redis OSS PFADD, PFCOUNT et PFMERGE. • Les requêtes de plage lexicographique avec les nouvelles commandes ZRANGEBYLEX, ZLEXCOUNT et ZREMRANGEBYLEX. • Plusieurs nouveaux correctifs, consistant notamment à empêcher un nœud primaire d'envoyer des données obsolètes vers des nœuds de réplica en mettant en échec la synchronisation principale quand un processus enfant (bgsave) d'enregistrement en arrière-plan s'arrête de manière inattendue. Pour plus d'informations sur HyperLogLog, voir la nouvelle structure de données de Redis OSS : le HyperLogLog. Pour plus d'informations sur PFADD, PFCOUNT et PFMERGE, consultez la documentation Redis OSS et cliquez sur. HyperLogLog	11 mars 2015
Prise en charge des balises de répartition des coûts	ElastiCache ajoute la prise en charge des balises de répartition des coûts. Pour de plus amples informations, veuillez consulter Surveillance des coûts avec des balises de répartition des coûts.	9 février 2015

Modification	Description	Date de modification
Support pour la AWS GovCloud région (ouest des États-Unis)	ElastiCache ajoute le support pour la région AWS GovCloud (US-Ouest) (us-gov-west-1).	29 janvier 2015
Prise en charge de la région Europe (Francfort)	ElastiCache ajoute le soutien à la région Europe (Francfort) (eu-central-1).	19 janvier 2015
Support multi-AZ pour les groupes de réplication Redis OSS	ElastiCache ajoute la prise en charge du multi-AZ depuis le nœud principal vers une réplique en lecture dans un groupe de réplication Redis OSS. ElastiCache surveille l'état du groupe de réplication. En cas de défaillance du serveur principal, transforme ElastiCache automatiquement un réplica en appareil principal, puis remplace le réplica. Pour de plus amples informations, veuillez consulter Minimiser les temps d'arrêt en ElastiCache utilisant le multi-AZ avec Valkey et Redis OSS.	24 octobre 2014
AWS CloudTrail enregistrement des appels d'API pris en charge	ElastiCache ajoute la prise en charge AWS CloudTrail de l'utilisation pour enregistrer tous les appels ElastiCache d'API. Pour de plus amples informations, veuillez consulter Journalisation des appels ElastiCache d'API Amazon avec AWS CloudTrail.	15 septembre 2014
Nouvelles tailles d'instance prises en charge	ElastiCache ajoute la prise en charge d'instances à usage général (T2) supplémentaires. Pour de plus amples informations, veuillez consulter Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres.	11 septembre 2014

Modification	Description	Date de modification
Placement de nœud flexible pris en charge pour Memcached	ElastiCache ajoute la prise en charge de la création de nœuds Memcached dans plusieurs zones de disponibilité.	23 juillet 2014
Nouvelles tailles d'instance prises en charge	ElastiCache ajoute la prise en charge d'instances à usage général (M3) supplémentaires et d'instances optimisées pour la mémoire (R3). Pour de plus amples informations, veuillez consulter Configuration des paramètres du moteur à l'aide de groupes de ElastiCache paramètres.	1 juillet 2014
Découverte automatique PHP	Ajouter la prise en charge de la découverte automatique PHP version 5.5.	13 mai 2014
Backup et restauration pour les clusters Redis OSS	Dans cette version, ElastiCache permet aux clients de créer des instantanés de leurs clusters Redis OSS et de créer de nouveaux clusters à l'aide de ces instantanés. Une sauvegarde est une copie du cluster à un moment précis. Elle comprend les métadonnées du cluster et toutes les données du cache Redis OSS. Les sauvegardes sont stockées dans Amazon S3 et les clients peuvent restaurer les données à partir d'un instantané dans un nouveau cluster à tout moment. Pour de plus amples informations, veuillez consulter Instantané et restauration.	24 avril 2014

Modification	Description	Date de modification
Redis OSS 2.8.6	ElastiCache supporte Redis OSS 2.8.6, en plus de Redis OSS 2.6.13. Avec Redis OSS 2.8.6, les clients peuvent améliorer la résilience et la tolérance aux pannes des répliques de lecture, grâce à la prise en charge de la resynchronisation partielle et à un nombre minimum de répliques de lecture défini par l'utilisateur qui doivent être disponibles à tout moment. Redis OSS 2.8.6 offre également un support complet pour publish-and-subscribe, où les clients peuvent être informés des événements qui se produisent sur le serveur.	13 mars 2014
Moteur de cache Redis OSS	ElastiCache propose le logiciel de moteur de cache Redis OSS, en plus de Memcached. Les clients qui utilisent actuellement Redis OSS peuvent « créer » un nouveau cluster de cache ElastiCache Redis OSS avec leurs données existantes provenant d'un fichier instantané Redis OSS, ce qui facilite la migration vers un environne ment géré. ElastiCache Pour prendre en charge les fonctionnalités de réplication de Redis OSS, l' ElastiCache API prend désormais en charge les groupes de réplication. Les clients peuvent créer un groupe de réplicati on avec un nœud de cache Redis OSS principal et ajouter un ou plusieurs nœuds de réplication de lecture qui restent automatiquement synchroni sés avec les données de cache du nœud principal . Les applications de lecture intensive peuvent être déchargées sur un réplica en lecture, réduisant ainsi la charge du nœud principal. Les réplicas en lecture peuvent également se prémunir contre la perte de données en cas de défaillance d'un nœud de cache principal.	3 septembre 2013

Modification	Description	Date de modification
Prise en charge d'Amazon Virtual Private Cloud (VPC) par défaut	Dans cette version, ElastiCache il est entièreme nt intégré à Amazon Virtual Private Cloud (VPC). Pour les nouveaux clients, les clusters de cache sont créés dans un Amazon VPC par défaut. Pour de plus amples informations, veuillez consulter Amazon VPCs et la ElastiCache sécurité.	8 janvier 2013
Prise en charge de PHP pour la découverte automatique du nœud de cache	La première version de découverte automatique du nœud de cache supportait les programmes Java. Dans cette version, ElastiCache intègre le support de découverte automatique des nœuds de cache à PHP.	2 janvier 2013
Prise en charge d'Amazon Virtual Private Cloud (VPC)	Dans cette version, les ElastiCache clusters peuvent être lancés dans Amazon Virtual Private Cloud (VPC). Par défaut, les clusters de cache des nouveaux clients sont créés automatiquement dans une instance Amazon VPC. Les clients existants peuvent migrer vers Amazon VPC à leur propre rythme. Pour de plus amples informations, veuillez consulter Amazon VPCs et la ElastiCache sécurité.	20 décembre 2012

Modification	Description	Date de modification
Découverte automatique des nœuds de cache et nouvelle version de moteur de cache	ElastiCache permet la découverte automatique des nœuds de cache, c'est-à-dire la capacité des programmes clients de déterminer automatiq uement tous les nœuds de cache d'un cluster, ainsi que d'établir et de maintenir des connexions avec tous ces nœuds. Cette édition offre également une nouvelle version de moteur de cache : Memcached version 1.4.14. Ce nouveau moteur de cache offre une capacité de rééquilibrage de section améliorée, des améliorations considérables au niveau de l'évoluti vité et des performances, ainsi que plusieurs correctifs. Plusieurs nouveaux paramètres de cache peuvent être configurés. Pour de plus amples informations, veuillez consulter Configura tion des paramètres du moteur à l'aide de groupes de ElastiCache paramètres.	28 novembre 2012
Nouveaux types de nœuds de cache	Cette version fournit quatre types de nœuds de cache supplémentaires.	13 novembre 2012
Nœuds de cache réservés	Cette version ajoute la prise en charge des nœuds de cache réservés. q	5 avril 2012
Nouveau guide	Il s'agit de la première version du guide de ElastiCache l'utilisateur Amazon.	22 août 2011

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

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.