



ElastiCache pour le guide de l'utilisateur de Memcached

Amazon ElastiCache



Version de l'API 2015-02-02

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ElastiCache: ElastiCache pour le guide de l'utilisateur de Memcached

Copyright © 2024 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 ElastiCache que Memcached ?	1
Mise en cache sans serveur	1
Clusters auto-conçus	2
Comment ça marche	2
Cache et moteurs de mise en cache	2
Cas d'utilisation	8
Stockage de données en mémoire	8
Témoignages de clients ElastiCache	10
Ressources ElastiCache for Memcached	10
Outils de gestion de votre implémentation	12
Utilisation de AWS Management Console	12
Utilisation de la AWS CLI	12
Utilisation de l'SDK AWS	12
Utilisation de l'API ElastiCache	12
Consultez aussi	13
Choix entre deux options de déploiement	13
Comparaison des caches auto-conçus Memcached et Redis	14
Mise en route avec ElastiCache for Memcached	21
Configuration	21
Inscrivez-vous pour un Compte AWS	21
Création d'un utilisateur doté d'un accès administratif	22
Octroi d'un accès par programmation	24
Configuration d'autorisations	25
Configuration d'EC2	26
Octroi d'un accès au réseau	27
Créer un cache	28
Création d'un cache sans serveur	28
Lire et écrire des données	29
(Facultatif) Nettoyer	34
Étapes suivantes	35
Didacticiel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un VPC Amazon	36
Étape 1 : Créer un cache ElastiCache	36
Étape 2 : créer une fonction Lambda	38

Étape 3 : Tester la fonction Lambda	43
Tutoriels et vidéos	44
Vidéos	45
Choix des régions et des zones de disponibilité	51
Considérations sur les zones de disponibilité	52
Régions et points de terminaison pris en charge	54
Localisation de vos nœuds	60
Utilisation de Local Zones	60
Activation d'une zone locale	61
Utilisation d'Outposts	62
Utilisation d'Outposts avec la console Memcached	62
Utilisation d'Outposts avec la CLI AWS	64
Conception de votre propre cluster ElastiCache	66
ElastiCache pour les composants et fonctionnalités de Memcached	67
Nœuds	67
Clusters	68
AWS Régions et zones de disponibilité	69
Points de terminaison	70
Groupes de paramètres	71
Sécurité	71
Groupes de sous-réseaux	71
Événements	72
Gestion des clusters	72
Choix d'un type de réseau	74
Mise à niveau des données	76
Découverte automatique	77
Préparation d'un cluster	121
Création d'un cluster	128
Affichage des détails d'un cluster	131
Modification d'un cluster	136
Redémarrage d'un cluster	140
Ajout de nœuds à un cluster	142
Suppression de nœuds d'un cluster	148
Annulation des opérations d'ajout ou de suppression de nœud en attente	154
Suppression d'un cluster	155
Accès à votre cluster	158

Recherche de points de terminaison de connexion	164
Gestion des nœuds	172
Affichage de l'état du ElastiCache nœud	172
Connexion aux nœuds	178
Types de nœuds pris en charge	181
Remplacement de nœuds	191
Nœuds réservés	194
Migration des nœuds de génération précédente	206
Travailler avec ElastiCache	208
Instantané et restauration	208
Constraints	209
Planification des sauvegardes automatiques	210
Réalisation de sauvegardes manuelles	211
Création d'une sauvegarde finale	213
Description des sauvegardes	215
Copie de sauvegardes	217
Restaurer à partir d'une sauvegarde	219
Suppression d'une sauvegarde	220
Étiquetage des sauvegardes	221
Versions du moteur et mise à niveau	222
Versions de Memcached prises en charge	223
Versions du moteur et mise à niveau	227
Comment mettre à niveau les versions de moteur	228
Bonnes pratiques et stratégies de mise en cache	230
Bonnes pratiques avec les clients Memcached	230
Commandes Memcached prises en charge	238
Stratégies de mise en cache	239
Gestion de votre cluster auto-conçu	245
Gestion de la maintenance	245
Configuration des paramètres de moteur à l'aide de groupes de paramètres	247
Mise à l'échelle ElastiCache pour Memcached	290
Mise à l'échelle ElastiCache pour Memcached	290
Définition de limites de mise à l'échelle pour gérer les coûts	290
Pré-dimensionnement avec Serverless ElastiCache	290
Définition des limites de dimensionnement à l'aide de la console et AWS CLI	292
Mise à l'échelle ElastiCache pour les clusters conçus par Memcached	293

Étiquetage de vos ressources ElastiCache	298
Surveillance des coûts avec des balises	305
Gestion des balises à l'aide du kit AWS CLI	307
Gestion des balises à l'aide de l'API ElastiCache	311
Cadre Amazon ElastiCache Well-Architected	313
Pilier Excellence opérationnelle	314
Pilier Sécurité	323
Pilier Fiabilité	330
Pilier Efficacité des performances	336
Pilier Optimisation des coûts	348
Résolution des problèmes	355
Problèmes de connexion	355
Erreurs du client Redis	356
Résolution des problèmes de latence élevée en mode ElastiCache Serverless	356
Résolution des problèmes de régulation dans Serverless ElastiCache	358
Rubriques connexes	359
Étapes de dépannage supplémentaires	360
Groupes de sécurité	360
Listes ACL réseau	361
Tables de routage	362
Résolution DNS	363
Identification des problèmes liés aux diagnostics côté serveur	363
Validation de connectivité réseau	369
Limites liées au réseau	372
Utilisation de l'UC	373
Connexions résiliées côté serveur	376
Dépannage côté client pour les instances Amazon EC2	378
Disséquer le temps nécessaire pour compléter une seule demande	379
Sécurité	382
Protection des données	383
Sécurité des données dans Amazon ElastiCache	384
Confidentialité du trafic inter-réseau	393
Sécurité des VPC Amazon et ElastiCache	393
API Amazon ElastiCache et points de terminaison d'un VPC d'interface (AWS PrivateLink) .	418
Sous-réseaux et groupes de sous-réseaux	421
Gestion de l'identité et des accès	430

Public ciblé	430
Authentification par des identités	431
Gestion des accès à l'aide de politiques	435
Comment Amazon ElastiCache travaille avec IAM	438
Exemples de politiques basées sur l'identité	446
Résolution des problèmes	449
Contrôle d'accès	451
Présentation de la gestion des accès	452
Validation de conformité	484
En savoir plus	486
Résilience	486
Atténuation des défaillances	487
Sécurité de l'infrastructure	489
Mises à jour de service	490
Gestion des mises à jour des services	490
Journalisation et surveillance	496
Métriques et événements pour les caches sans serveur	496
Métriques pour les caches sans serveur	496
Événements pour les caches sans serveur	500
Métriques et événements des clusters auto-conçus	505
Métriques pour les clusters auto-conçus	506
Événements pour les clusters auto-conçus	506
Utilisation de la surveillance	514
Surveillance des événements avec Amazon SNS	529
Journalisation des appels d'API Amazon ElastiCache avec AWS CloudTrail	547
Informations Amazon ElastiCache dans CloudTrail	547
Présentation des entrées des fichiers journaux Amazon ElastiCache	548
Quotas	553
Référence	554
Utilisation de l'API ElastiCache	554
Utilisation de l'API Query	554
Bibliothèques disponibles	558
Applications de dépannage	558
Configuration de la AWS CLI pour ElastiCache	559
Prérequis	560
Téléchargement des outils de ligne de commande	562

Installation des outils	562
Indication des informations d'identification pour les outils	563
Variables d'environnement	564
Messages d'erreur	565
Notifications	567
Notifications générales ElastiCache	567
Notifications ElastiCache for Memcached	567
.....	569
AWS Glossaire	587
.....	dlxxxviii

Qu'est-ce qu'Amazon ElastiCache pour Memcached ?

Bienvenue dans le guide de l'utilisateur ElastiCache d'Amazon pour Memcached. 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 à la fois avec les moteurs Redis et Memcached. Utilisez le guide pour le moteur qui vous intéresse. Si vous n'êtes pas sûr du moteur que vous souhaitez utiliser, consultez [Comparaison des caches auto-conçus Memcached et Redis](#) dans ce guide.

Mise en cache sans serveur

ElastiCache for Memcached propose une mise en cache sans serveur, qui simplifie l'ajout et l'exploitation d'un cache basé sur Memcached pour votre application. ElastiCache for Memcached 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 et le calcul utilisés par votre application, et adapte automatiquement la capacité 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 et du logiciel sous-jacents. 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 pour Memcached Serverless est compatible avec Memcached 1.6 et versions ultérieures.

Conception de votre propre cluster ElastiCache pour Memcached

Si vous avez besoin d'un contrôle précis de votre ElastiCache cluster Memcached, vous pouvez choisir de concevoir votre propre cluster Memcached avec. 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 disponibilité de votre cluster. Comme il 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 cluster ElastiCache pour Memcached 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 entre plusieurs zones de disponibilité, en fonction de vos besoins. 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 à quel moment appliquer les nouveaux correctifs logiciels à votre cluster Memcached.

Lorsque vous concevez le vôtre ElastiCache pour Memcached, vous pouvez choisir d'exécuter Memcached 1.4 ou version ultérieure.

Fonctionnement

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

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 for Memcached 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 à la fois avec les moteurs Redis et Memcached. Utilisez le guide pour le moteur qui vous intéresse. Si vous n'êtes pas sûr du moteur que vous souhaitez utiliser, consultez [Comparaison des caches auto-conçus Memcached et Redis](#) dans ce guide.

Rubriques

- [Comment ElastiCache fonctionne Memcached](#)
- [Grille tarifaire](#)

Comment ElastiCache fonctionne Memcached

ElastiCache pour Memcached Serverless

ElastiCache for Memcached 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 d'attribuer un nom à votre cache et vous recevez un point de terminaison unique que vous pouvez configurer dans votre client Memcached pour commencer à accéder à votre cache.

Note

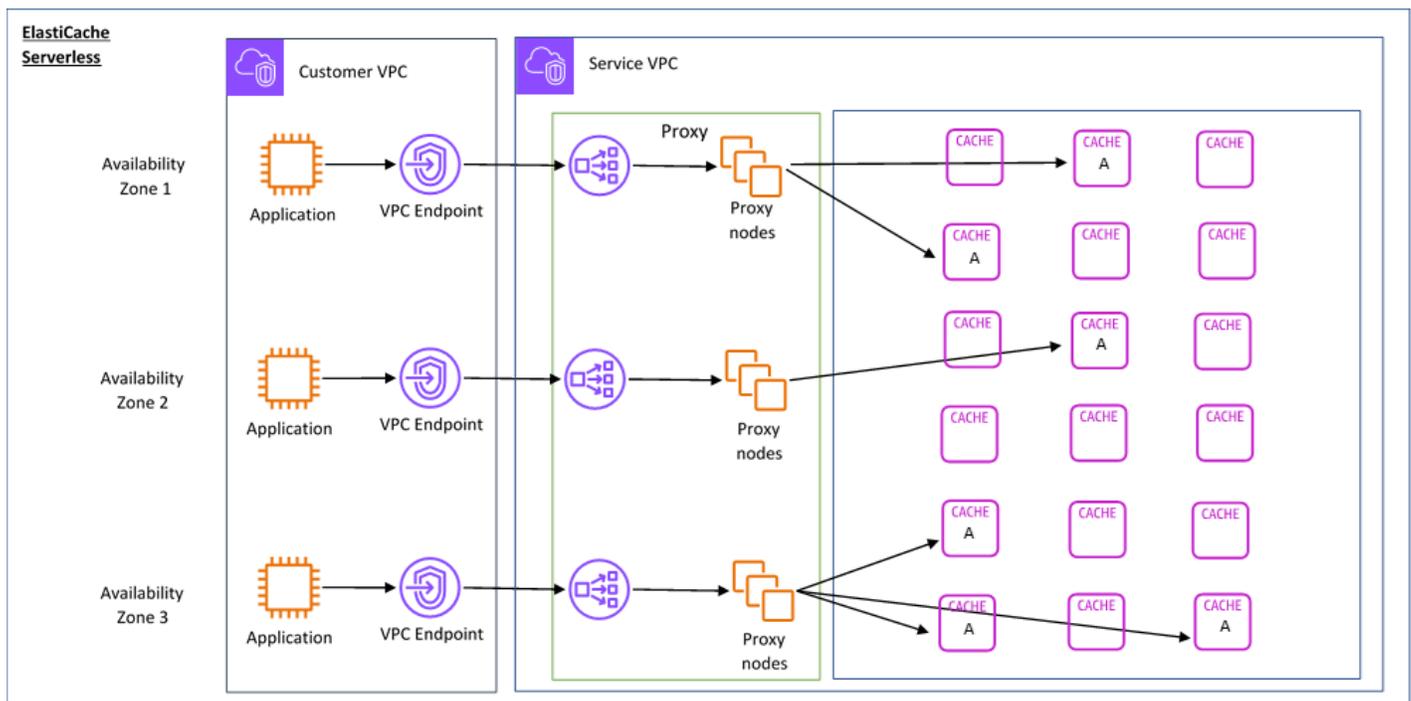
ElastiCache pour Memcached Serverless n'est compatible qu'avec les clients Memcached compatibles avec le protocole TLS.

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. veuillez consulter [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 de Memcached 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.

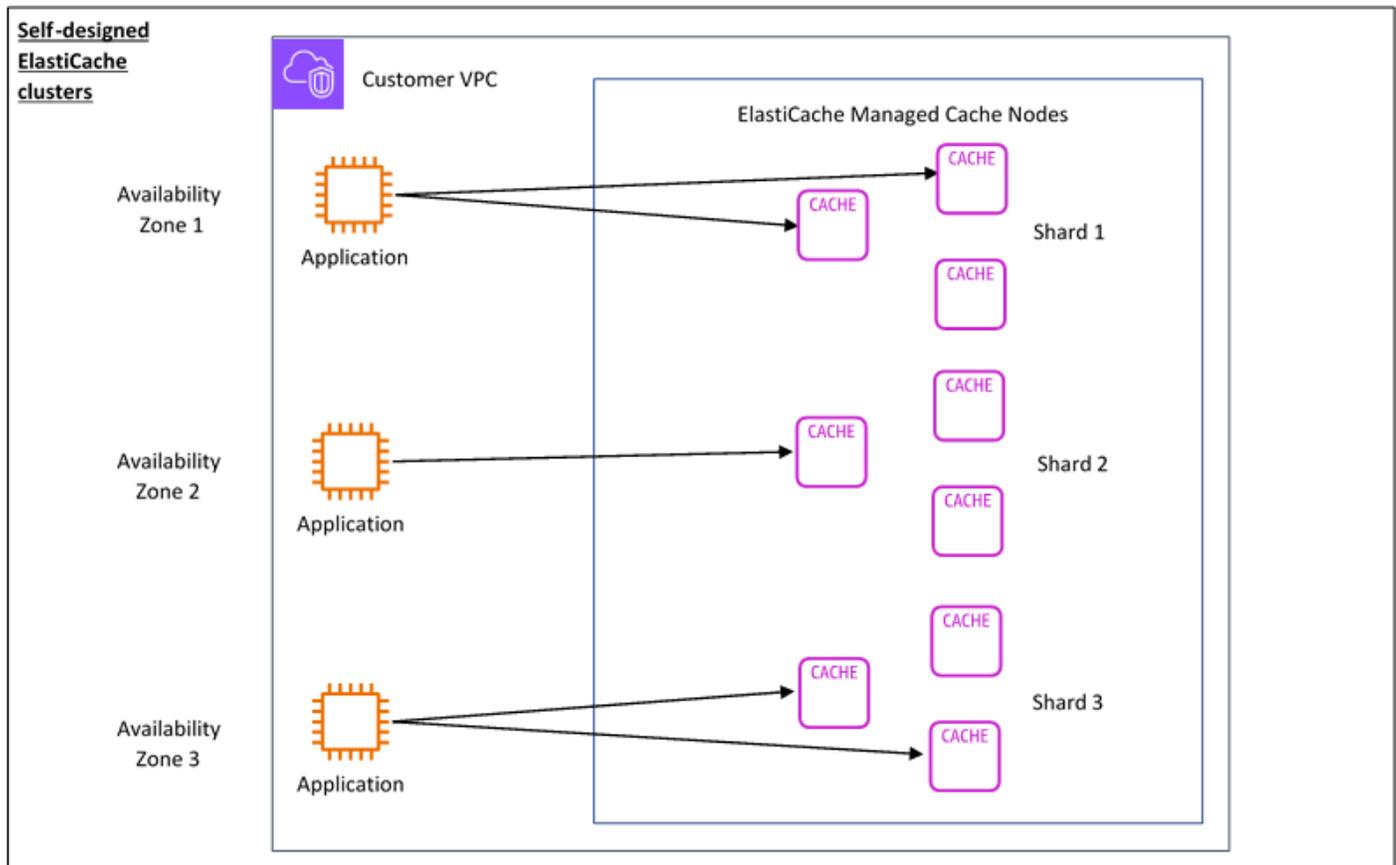
Clusters conçus ElastiCache par vos soins

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 permet de contrôler plus précisément la configuration et la mise à l'échelle du cluster.

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 zones de disponibilité pour garantir 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 [Configuration des paramètres de moteur à l'aide de groupes de paramètres](#) pour configurer le moteur du 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 des nœuds.

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 le vôtre ElastiCache pour les clusters Memcached, 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 (eCPU)

Vous payez pour les requêtes que votre application exécute sur des unités de ElastiCache traitement ElastiCache sans serveur (eCPU), une unité qui inclut à la fois le temps passé par le 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 demande SET qui transfère 3,2 Ko de données consommera 3,2 ECPU.
- Les commandes qui s'exécutent sur plusieurs éléments consommeront proportionnellement plus d'ECPU. Par exemple, si votre application effectue une opération de lecture multiple sur 3 éléments, elle consommera 3 ECPU.
- Les commandes qui s'exécutent sur un plus grand nombre d'éléments et qui transfèrent davantage de données consomment des ECPU 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, si elle ne transfère que 2 Ko de données, elle consommera 3 ECPU.

ElastiCache Serverless émet une nouvelle métrique appelée `ElastiCacheProcessingUnits` qui vous aide à comprendre les eCPU consommés par votre charge de travail.

Heures d'utilisation des nœuds

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

Cas d'utilisation courants d'ElastiCache et ce que peut vous apporter ElastiCache

Qu'il s'agisse de fournir les dernières actualités ou 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 \(pour les utilisateurs Web impatientes, patienter le temps d'un clignement d'œil est encore trop long\)](#) », 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 indiquent certains moyens à l'aide d'Elasticache qui peuvent améliorer les performances globales de votre application.

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 plus d'informations, consultez les ressources suivantes :

- [Stratégies de mise en cache](#) dans le Guide de l'utilisateur d'ElastiCache pour Memcached

Témoignages de clients ElastiCache

Pour en savoir plus sur la façon dont les entreprises comme Airbnb, PBS, Esri et d'autres tirent parti d'Amazon ElastiCache pour développer leur activité et améliorer l'expérience client, veuillez consulter [Comment d'autres personnes utilisent Amazon ElastiCache](#).

Vous pouvez également regarder les [didacticiels vidéo](#) pour obtenir des cas d'utilisation utilisateur d'ElastiCache supplémentaires.

Ressources ElastiCache for Memcached

Nous recommandons de commencer par lire les sections suivantes et de vous y référer dès que vous en avez besoin:

- Description des tarifs et services – La [page de détails des produits](#) fournit une description générale des produits ElastiCache, une description des services et les tarifs applicables.
- Vidéos ElastiCache – La section [Vidéos ElastiCache](#) comporte des vidéos qui présentent Amazon ElastiCache for Memcached, couvrent les cas d'utilisation courants et expliquent comment utiliser ElastiCache for Memcached pour réduire la latence et améliorer le débit de vos applications.
- Mise en route – La section [Mise en route avec Amazon ElastiCache for Memcached](#) comprend un exemple qui vous guide pas à pas pour créer un cluster de cache, pour autoriser l'accès au cluster de cache, pour se connecter au nœud de cache et supprimer le cluster de cache.
- Performances évolutives – Le livre blanc [Performance at Scale with Amazon ElastiCache](#) aborde les stratégies de mise en cache qui permettent à votre application de fonctionner à l'échelle.

Si vous souhaitez utiliser l'AWS Command Line Interface (AWS CLI), ces documents peuvent vous aider à vous lancer :

- [Documentation AWS Command Line Interface](#)

Cette section fournit des informations sur le téléchargement et la mise en place de l'AWS CLI, la mise en place de l'AWS CLI dans votre système, ainsi que la définition de vos informations d'identification AWS.

- [Documentation AWS CLI pour ElastiCache](#)

Ceci est un document distinct qui couvre toutes les commandes AWS CLI pour ElastiCache, y compris leur syntaxe et des exemples.

Vous pouvez développer des applications pour utiliser l'API ElastiCache avec différents langages de programmation courants. Voici quelques ressources :

- [Outils pour Amazon Web Services](#)

Amazon Web Services fournit plusieurs kits de développement logiciel (SDK) qui prennent en charge ElastiCache for Memcached. Vous pouvez programmer pour ElastiCache en utilisant Java, .NET, PHP, Ruby, ainsi que d'autres langages. Ces kits de développement logiciel peuvent simplifier considérablement le développement de vos applications en formatant vos requêtes à ElastiCache pour analyser syntaxiquement les réponses et fournir des logiques de nouvelle tentative et une gestion des erreurs.

- [Utilisation de l'API ElastiCache](#)

Si vous ne voulez pas utiliser les kits de développement logiciel AWS, vous pouvez interagir avec ElastiCache directement en utilisant 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 d'API Amazon ElastiCache](#)

Il s'agit d'un document distinct couvrant toutes les opérations de l'API ElastiCache, y compris leur syntaxe et des exemples.

Outils de gestion de votre implémentation

Une fois que vous avez accordé à votre instance Amazon EC2 l'accès à votre cluster ElastiCache, vous disposez de quatre moyens pour gérer votre cluster ElastiCache : la AWS Management Console, la AWS CLI pour ElastiCache, le SDK AWS pour ElastiCache et l'API ElastiCache.

Utilisation de AWS Management Console

La AWS Management Console est le moyen le plus simple de gérer Amazon ElastiCache for Memcached. La console vous permet de créer des clusters de cache, d'ajouter et de supprimer des nœuds de cache et d'effectuer d'autres tâches administratives sans avoir à écrire des lignes de code. La console fournit également une représentation graphique de la performance des nœuds de cache à partir de CloudWatch, en affichant l'activité du moteur de cache, la mémoire et l'utilisation de l'UC et d'autres métriques. Pour plus d'informations, consultez les rubriques du Guide de l'utilisateur.

Utilisation de la AWS CLI

Vous pouvez également utiliser la AWS Command Line Interface (AWS CLI) pour ElastiCache. La AWS CLI vous permet d'exécuter facilement des opérations une par une, par exemple démarrer ou arrêter votre cluster de cache. Vous pouvez également appeler la commande AWS CLI pour ElastiCache en utilisant le langage de scripting de votre choix, afin d'automatiser des tâches répétitives. Pour plus d'informations sur la AWS CLI, consultez le Guide de l'utilisateur et la [Référence des commandes AWS CLI](#).

Utilisation de l'SDK AWS

Pour accéder à ElastiCache à partir d'une application, vous pouvez utiliser l'un des kits de développement logiciel AWS (SDK). Les SDKs encapsulent les appels de l'API ElastiCache et protègent vos applications des détails de bas niveau de l'API ElastiCache. Vous fournissez vos informations d'identification et les bibliothèques SDK prennent en charge l'authentification et la signature des demandes. Pour de plus amples informations sur l'utilisation des kits SDK AWS, veuillez consulter [Outils pour Amazon Web Services](#).

Utilisation de l'API ElastiCache

Vous pouvez également écrire du code d'application directement contre l'API du service web ElastiCache. Lorsque vous utilisez l'API, vous devez écrire le code nécessaire à l'élaboration et à

l'authentification de vos requêtes HTTP, à l'interprétation des résultats d'ElastiCache et à la prise en charge d'erreurs éventuelles. Pour plus d'informations sur l'API, consultez [Utilisation de l'API ElastiCache](#).

Consultez aussi

Pour des informations plus détaillées sur la gestion de votre déploiement d'Amazon ElastiCache for Memcached, consultez ce qui suit :

- [Travailler avec ElastiCache](#)
- [Confidentialité du trafic inter-réseau](#)
- [Journalisation et surveillance dans Amazon ElastiCache](#)

Choix entre deux options de déploiement

Amazon ElastiCache propose deux options de déploiement :

- Mise en cache sans serveur
- Conception de votre propre cluster

Mise en cache sans serveur

Amazon ElastiCache sans serveur simplifie la création de cache. Il est mis à l'échelle instantanément pour prendre en charge les applications les plus exigeantes des clients. Avec ElastiCache sans serveur, vous pouvez créer un cache hautement disponible et évolutif en moins d'une minute, sans avoir besoin de provisionner, de planifier et de gérer la capacité du cluster de cache. ElastiCache sans serveur stocke automatiquement les données de manière redondante dans trois zones de disponibilité et fournit un [contrat de niveau de service \(SLA\)](#) de disponibilité de 99,99 %. ElastiCache assure la réplication automatique des données entre les zones de disponibilité sans que vous ayez à gérer manuellement les réplicas et le logiciel personnalisé pour qu'elles restent synchronisées.

Clusters auto-conçus

Si vous avez besoin de contrôler votre cluster ElastiCache for Memcached avec précision, vous pouvez choisir de concevoir votre propre cluster Memcached avec ElastiCache. ElastiCache vous permet d'exploiter un cluster basé sur un nœud, en choisissant le type de nœud, le nombre de nœuds et le placement des nœuds dans les zones de disponibilité AWS de votre cluster. Étant donné

qu'ElastiCache est un service entièrement géré, il gère automatiquement le provisionnement du matériel, la surveillance, le remplacement des nœuds et les correctifs logiciels pour votre cluster.

Choix entre deux options de déploiement

Choisissez la mise en cache sans serveur si :

- Vous créez un cache pour des charges de travail nouvelles ou inconnues.
- Le trafic de votre application est imprévisible.
- Vous cherchez la méthode la plus simple pour démarrer avec un cache.

Choisissez de concevoir votre propre cluster ElastiCache si :

- Vous utilisez déjà ElastiCache sans serveur et vous souhaitez contrôler plus précisément le type de nœud qui exécute Memcached, le nombre de nœuds et le placement des nœuds.
- Vous ne pensez pas que le trafic de votre application fluctue beaucoup ou vous pouvez prévoir avec précision les pics et les baisses de trafic de votre application.
- Vous pouvez prévoir vos exigences de capacité pour contrôler les coûts.

Voir aussi:

- [Conception et gestion de votre propre cluster ElastiCache pour l'implémentation de Memcached](#)

Comparaison des caches auto-conçus Memcached et Redis

Amazon ElastiCache prend en charge les moteurs de cache Memcached et Redis. 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 Redis avec une version de ElastiCache for Redis si les conditions suivantes s'appliquent à vous :

- ElastiCache pour Redis version 7.0 (améliorée)

Vous souhaitez utiliser les [Fonctions Redis](#), le [Pub/Sub partitionné](#) ou les [Améliorations des ACL Redis](#). Pour plus d'informations, consultez la rubrique [Redis version 7.0 \(améliorée\)](#).

- ElastiCache pour Redis version 6.2 (améliorée)

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 pour Redis version 6.0 (améliorée)

Vous souhaitez authentifier les utilisateurs avec le contrôle d'accès basé sur les rôles.

Pour plus d'informations, consultez la rubrique [Redis version 6.0 \(améliorée\)](#).

- ElastiCache pour Redis version 5.0.0 (améliorée)

Vous souhaitez utiliser les [flux Redis](#), une structure de données de journal qui permet aux producteurs d'ajouter de nouveaux éléments en temps réel. Elle permet également aux consommateurs d'utiliser des messages en mode bloquant ou non bloquant.

Pour plus d'informations, consultez [Redis version 5.0.0 \(améliorée\)](#).

- ElastiCache pour Redis version 4.0.10 (améliorée)

Prend en charge le chiffrement et l'ajout ou la suppression dynamiques des partitions à partir de votre cluster Redis (mode cluster activé).

Pour plus d'informations, consultez [Redis version 4.0.10 \(améliorée\)](#).

Les versions suivantes sont obsolètes, ont atteint ou atteindront bientôt leur fin de vie.

- ElastiCache pour Redis version 3.2.10 (améliorée)

Prend en charge la capacité à ajouter ou supprimer de manière dynamique des partitions de votre cluster Redis (mode cluster activé).

 Important

Actuellement ElastiCache , Redis 3.2.10 ne prend pas en charge le chiffrement.

Pour plus d'informations, consultez les ressources suivantes :

- [Redis Version 3.2.10 \(améliorée\)](#)
- Pour accéder aux bonnes pratiques de repartitionnement en ligne pour Redis, veuillez consulter les sections suivantes :
 - [Bonnes pratiques : repartitionnement en ligne](#)
 - [Repartitionnement en ligne et rééquilibrage de la partition pour Redis \(mode cluster activé\)](#)
- Pour plus d'informations sur le dimensionnement de Redis, consultez [Dimensionnement](#).

- ElastiCache pour Redis version 3.2.6 (améliorée)

Si vous avez besoin des fonctionnalités des versions antérieures de Redis ainsi que des fonctionnalités suivantes, optez ElastiCache pour Redis 3.2.6 :

- Chiffrement en transit. Pour plus d'informations, consultez [Amazon ElastiCache pour le chiffrement Redis In-Transit](#).
- Chiffrement au repos. Pour plus d'informations, consultez [Amazon ElastiCache for Redis At-Rest Encryption](#).
- ElastiCache pour Redis (mode cluster activé) version 3.2.4

Si vous avez besoin de toutes les fonctionnalités de Redis 2.8.x en plus des fonctionnalités suivantes, choisissez Redis 3.2.4 (mode cluster) :

- 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 pour Redis (mode non clusterisé) 2.8.x et 3.2.4 (amélioré)

Si les éléments suivants s'appliquent à votre situation, choisissez Redis 2.8.x ou Redis 3.2.4 (mode non-cluster) :

- 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 avez 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 de Memcached, Redis (mode cluster désactivé) et Redis (mode cluster activé)

	Memcached	Redis (Mode cluster désactivé)	Redis (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	Redis (Mode cluster désactivé)	Redis (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

Redis (Mode cluster
désactivé)Redis (mode cluster
activé)

+ Exclut les versions obsolètes, arrivées en fin de vie ou sur le point d'atteindre leur fin de vie.

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 plus d'informations, consultez [Supporté ElastiCache pour les versions Memcached](#) ou [Supporté ElastiCache pour les versions Redis](#).

Mise en route avec Amazon ElastiCache for Memcached

Les rubriques de cette section vous guident dans le processus de création, d'octroi d'accès, de connexion et de suppression d'un cache sans serveur Memcached à l'aide de la console ElastiCache.

Rubriques

- [Configuration](#)
- [Étape 1 : Créer un cache](#)
- [Étape 2 : Lire et écrire des données dans le cache](#)
- [Étape 3 : \(Facultatif\) Nettoyer](#)
- [Étapes suivantes](#)
- [Didacticiel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un VPC Amazon](#)
- [Tutoriels et vidéos ElastiCache](#)

Configuration

Pour configurer ElastiCache :

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 \(nouveaux ElastiCache utilisateurs uniquement\)](#)
- [Configuration d'EC2](#)
- [Octroi d'un accès au réseau à votre cache à partir d'un groupe de sécurité Amazon VPC](#)

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez l'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 Console](#) tant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez [Connexion en tant qu'utilisateur racine](#) dans le Guide de l'utilisateur Connexion à AWS .

2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

Pour obtenir des instructions, consultez [Activation d' AWS IAM Identity Center](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

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

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

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

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

Attribuer l'accès à des utilisateurs supplémentaires

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

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

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

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

Octroi d'un accès par programmation

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

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

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées aux AWS CLI AWS SDK ou AWS aux API.	Suivez les instructions de l'interface que vous souhaitez utiliser. <ul style="list-style-type: none"> • Pour le AWS CLI, voir Configuration du AWS CLI à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur. • Pour les AWS SDK, les outils et les AWS API, consultez la section Authentification IAM Identity Center dans le Guide de référence AWS des SDK et des outils.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées aux AWS CLI AWS SDK ou AWS aux API.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer les AWS CLI demandes programmatiques adressées aux AWS SDK ou AWS aux API.	Suivez les instructions de l'interface que vous souhaitez utiliser. <ul style="list-style-type: none">• Pour le AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'AWS Command Line Interface utilisateur.• Pour les AWS SDK et les outils, voir Authentifier à l'aide d'informations d'identification à long terme dans le Guide de AWS référence des SDK et des outils.• Pour les AWS API, consultez la section Gestion des clés d'accès pour les utilisateurs IAM dans le guide de l'utilisateur IAM.

Voir aussi:

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

Configurez vos autorisations (nouveaux 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. Pour plus d'informations, voir la rubrique [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) du Guide de l'utilisateur IAM.

- Utilisateurs IAM :

- Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique [Création d'un rôle pour un utilisateur IAM](#) du Guide de l'utilisateur IAM.
- (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique [Ajout d'autorisations à un utilisateur \(console\)](#) du Guide de l'utilisateur IAM.

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 êtes autorisé à appeler `iam:createServiceLinkedRole` ou que vous avez créé le rôle ElastiCache lié au service.

Pour plus d'informations, consultez les ressources suivantes :

- [Création d'une politique](#) (IAM)
- [Politiques gérées par AWS pour Amazon ElastiCache](#)
- [Utilisation des rôles liés à un service pour Amazon ElastiCache](#)

Configuration d'EC2

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

- Si vous ne possédez pas encore d'instance EC2, découvrez comment configurer une instance EC2 en consultant le didacticiel [Démarez avec EC2](#).

- Votre instance EC2 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, vérifiez que votre instance EC2 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 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 instance EC2, votre groupe de sécurité doit autoriser l'accès à ces ports.

1. Connectez-vous à la console [Amazon EC2 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. Dans la zone Plage de ports, entrez 11211.
 - e. Dans la zone Source, choisissez N'importe où qui a la plage de ports (0.0.0.0/0) afin que toute instance Amazon EC2 que vous lancez dans votre réseau 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. Dans la zone Plage de ports, entrez 11212.
 - i. Dans la zone Source, choisissez N'importe où qui a la plage de ports (0.0.0.0/0) afin que toute instance Amazon EC2 que vous lancez dans votre réseau Amazon VPC puisse se connecter à votre cache.
 - j. Choisissez Enregistrer.

Étape 1 : Créer un cache

Le cache que vous êtes sur le point de lancer sera opérationnel et ne sera pas exécuté dans un environnement de test (sandbox). Vous devrez payer les frais d'utilisation standard d'ElastiCache pour le cache jusqu'à ce que vous le supprimiez. Le total des frais est minime (généralement moins d'un dollar) si vous terminez l'exercice décrit ici en une seule fois et que vous supprimez le cache lorsque vous avez terminé. Pour de plus amples informations sur les taux d'utilisation d'ElastiCache, veuillez consulter [Amazon ElastiCache](#).

Création d'un cache sans serveur

AWS Management Console

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

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 cache à l'aide d'AWS CLI

L'exemple AWS CLI suivant crée un 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 qu'ElastiCache a terminé de créer le cache, utilisez la commande `describe-serverless-caches`.

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'[Étape 2 : Lire et écrire des données dans le cache](#).

Étape 2 : Lire et écrire des données dans le cache

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

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

Configuration

Avant de commencer, assurez-vous que vous disposez des ports d'accès appropriés.

Port principal : 11211

Port optimisé pour la lecture : 11212

Les caches Memcached sans serveur annoncent deux ports portant le même nom d'hôte. Le port principal permet les écritures et les lectures avec les mêmes garanties de cohérence que OSS

Memcached. Le port optimisé pour la lecture permet les écritures et permet en outre de réduire la latence pour des lectures finalement cohérentes.

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

Pour en savoir plus sur la connexion à l'aide du client Java Memcached, consultez [ElastiCache chiffrement en transit \(TLS\)](#).

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)

Consultez 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"
```

Connexion à l'aide du client NodeJS/TS Memcached (Electrode-IO memcache)

Consultez <https://github.com/electrode-io/memcache> et <https://www.npmjs.com/package/memcache-client>

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`)

Consultez <https://crates.io/crates/memcache> et <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`)

Consultez <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`)

Consultez <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

Voir <https://github.com/cnblogs/EnyimMemcachedCore>

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

Vous pouvez maintenant passer à l'[Étape 3 : \(Facultatif\) Nettoyer](#).

Étape 3 : (Facultatif) Nettoyer

Utilisation de la 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 AWS Management Console et ouvrez la console Amazon ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Dans le tableau de bord de la console ElastiCache, sélectionnez le moteur exécuté par le cache à 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 dans la console ElastiCache. 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.

Utilisation d'AWS CLI

Le code suivant supprime le cache my-cache.

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

L'action d'interface de ligne de commande delete-serverless-cache supprime uniquement un cache sans serveur. Pour supprimer plusieurs caches, appelez delete-serverless-cache pour 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 la rubrique [ElastiCache delete-serverless-cache](#) dans AWS CLI.

Vous pouvez maintenant passer à l'[Étapes suivantes](#).

Étapes suivantes

Pour plus d'informations sur ElastiCache, consultez :

- [Travailler avec ElastiCache](#)
- [Mise à l'échelle ElastiCache pour Memcached](#)
- [Quotas pour ElastiCache](#)
- [ElastiCache meilleures pratiques et stratégies de mise en cache](#)
- [Affichage des événements ElastiCache](#)

Didacticiel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un VPC Amazon

Dans ce didacticiel, vous effectuez les opérations suivantes :

- Créez un cache Amazon ElastiCache dans votre réseau Amazon Virtual Private Cloud (Amazon VPC) par défaut, dans la région us-east-1.
- Créez une fonction Lambda pour accéder au cache ElastiCache. Lorsque vous créez la fonction Lambda, vous fournissez des ID de sous-réseau dans votre VPC Amazon, ainsi qu'un groupe de sécurité de VPC pour permettre à la fonction Lambda d'accéder aux ressources dans votre VPC. A titre d'illustration dans ce didacticiel, la fonction Lambda génère un UUID, l'écrit dans le cache et l'extrait du cache.
- Invoquez la fonction Lambda manuellement et vérifiez qu'elle a accédé au cache ElastiCache dans votre VPC.

Important

Ce didacticiel utilise le VPC Amazon par défaut dans la région us-east-1 de votre compte. Pour plus d'informations sur Amazon VPC, consultez [Mise en route sur Amazon VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Rubriques

- [Étape 1 : Créer un cache ElastiCache](#)
- [Étape 2 : créer une fonction Lambda](#)
- [Étape 3 : Tester la fonction Lambda](#)

Mise en route

[Étape 1 : Créer un cache ElastiCache](#)

Étape 1 : Créer un cache ElastiCache

Au cours de cette étape, vous allez créer un cache Amazon ElastiCache dans le réseau Amazon Virtual Private Cloud par défaut, dans la région us-east-1 de votre compte, à l'aide de l'interface de

ligne de commande AWS. Pour en savoir plus sur la création de caches sans serveur ElastiCache à l'aide de la console ou de l'API ElastiCache, consultez [Création d'un cluster](#) dans le Guide de l'utilisateur ElastiCache for Memcached.

AWS Management Console

Exécutez la commande AWS CLI suivante pour créer un cache sans serveur de cluster Memcached dans le VPC par défaut, dans la région us-east-1.

Linux

```
aws elasticache create-serverless-cache \  
--serverless-cache-name serverlessCacheForLambda \  
--region us-east-1 \  
--engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^  
--serverless-cache-name serverlessCacheForLambda ^  
--region us-east-1 ^  
--engine memcached
```

Notez que la valeur du champ Statut est définie sur CREATING. ElastiCache peut mettre plusieurs minutes à terminer la création de votre cluster.

Pour vérifier qu'ElastiCache a terminé de créer le cache, utilisez la commande `describe-serverless-caches`.

Linux

```
aws elasticache describe-serverless-caches \  
--serverless-cache-name serverlessCacheforLambda \  
--region us-east-1
```

Windows

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

```
--region us-east-1
```

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.

Une fois le nouveau cache créé, passez à l'[Étape 2 : créer une fonction Lambda](#).

Étape suivante :

[Étape 2 : créer une fonction Lambda](#)

Étape 2 : créer une fonction Lambda

À cette étape, vous effectuez les opérations suivantes :

1. Créez le package de déploiement d'une fonction Lambda avec l'exemple de code fourni.
2. Créez un rôle IAM (rôle d'exécution). Lorsque vous chargez le package de déploiement, vous devez spécifier ce rôle pour que Lambda puisse l'assumer et exécuter la fonction en votre nom. La stratégie d'autorisations permet aux autorisations AWS Lambda de configurer des interfaces réseau Elastic, ou ENI, de sorte que votre fonction Lambda puisse accéder aux ressources du VPC. Dans cet exemple, la fonction Lambda accède à un cluster ElastiCache dans le VPC.
3. Créez la fonction Lambda en chargeant le package de déploiement.

Étape suivante

[Étape 2.1 : Créer le package de déploiement](#)

Étape 2.1 : Créer le package de déploiement

Actuellement, l'exemple de code pour la fonction Lambda n'est fourni qu'en Python.

Python

L'exemple suivant de code Python lit et écrit un élément dans votre cluster ElastiCache. Copiez le code et enregistrez-le dans un fichier nommé `app.py`. Assurez-vous de remplacer la valeur `elasticache_config_endpoint` dans le code par l'adresse du point de terminaison que vous avez copiée à l'étape 1.

```
import uuid
import ssl
```

```
from pymemcache.client.base import Client

elasticache_config_endpoint = "serverlesscacheforlambda-
ces85m.serverless.use1.cache.amazonaws.com"
target_port = 11211

context = ssl.create_default_context()

memcached_client = Client((elasticache_config_endpoint, target_port),
    tls_context=context)

def lambda_handler(event, context):

    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex

    # put the UUID to the cache
    memcached_client.set("uuid", uuid_in, expire=500, noreply=False)

    # get the item (UUID) from the cache
    result = memcached_client.get("uuid")
    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 Memcached.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

    return "Fetched value from Memcached"
```

Ce code utilise la bibliothèque Python [pymemcache](#) pour placer des éléments dans votre cache et les récupérer. Pour créer un package de déploiement contenant pymemcache, suivez les étapes ci-dessous.

1. Dans le répertoire de votre projet contenant le fichier de code source `app.py`, créez un dossier package dans lequel installer la bibliothèque pymemcache.

```
mkdir package
```

2. Installez pymemcache à l'aide de pip.

```
pip install --target ./package pymemcache
```

3. Créez un fichier .zip contenant la bibliothèque pymemcache. Sous Linux et macOS, exécutez la commande suivante. Sous Windows, utilisez votre utilitaire zip préféré pour créer un fichier .zip avec la bibliothèque pymemache à la racine.

```
cd package
zip -r ../my_deployment_package.zip .
```

4. Ajoutez votre code de fonction dans le fichier .zip. Sous Linux et macOS, exécutez la commande suivante. Sous Windows, utilisez votre utilitaire zip préféré pour ajouter app.py à la racine de votre fichier .zip.

```
cd ..
zip my_deployment_package.zip app.py
```

Étape suivante

[Étape 2.2 : Créer un rôle IAM \(rôle d'exécution\)](#)

Étape 2.2 : Créer un rôle IAM (rôle d'exécution)

À cette étape, vous créez un rôle Identity and Access Management (IAM) AWS à l'aide du type de rôle prédéfini et de la stratégie d'accès ci-dessous :

- Fonction du service AWS de type AWS Lambda : ce rôle accorde les autorisations AWS Lambda requises pour assumer ce rôle.
- AWSLambdaVPCLambdaAccessExecutionRole : la stratégie d'autorisations d'accès que vous associez au rôle. Cette stratégie accorde les autorisations requises pour les actions EC2 dont AWS Lambda a besoin pour gérer les ENI. Vous pouvez consulter cette stratégie gérée par AWS dans la console IAM.

Pour en savoir plus sur les rôles utilisateur IAM, consultez la section [Rôles \(délégation et fédération\)](#) dans le Guide de l'utilisateur IAM.

Utilisez la procédure suivante pour créer le rôle IAM.

Pour créer un rôle IAM (d'exécution)

1. Connectez-vous à la console de gestion AWS et ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Choisissez Roles (Rôles), puis Create role (Créer un rôle).
 - Sous Type d'entité approuvée, choisissez Service AWS, puis sous Cas d'utilisation, choisissez Lambda. Cette opération accorde au service AWS Lambda les autorisations requises pour assumer ce rôle. Choisissez Next (Suivant).
 - Sous Ajouter des autorisations, recherchez **AWSLambdaVPCAccessExecutionRole** et cochez la case en regard de la politique.
 - Choisissez Next (Suivant).
 - Dans Role Name (Nom de rôle), utilisez un nom unique au sein de votre compte AWS (par exemple, lambda-vpc-execution-role).
 - Sélectionnez Create role (Créer un rôle).
3. Copiez l'ARN de rôle. Vous en aurez besoin à l'étape suivante lorsque vous créerez votre fonction Lambda.

Étape suivante

[Étape 2.3 : Charger le package de déploiement \(créer la fonction Lambda\)](#)

Étape 2.3 : Charger le package de déploiement (créer la fonction Lambda)

À cette étape, vous créez la fonction Lambda (AccessMemcached) à l'aide de la commande `create-fonction` de l'interface de ligne de commande AWS.

Dans le répertoire du projet contenant le fichier `.zip` de votre package de déploiement, exécutez la commande d'interface de ligne de commande Lambda `create-fonction` suivante.

Pour l'option `role`, utilisez l'ARN du rôle d'exécution que vous avez créé à l'étape 2.2. Pour `vpc-config`, entrez les listes des sous-réseaux de votre VPC par défaut, séparés par des virgules, et l'ID de 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 Vos VPC, puis choisissez le VPC par défaut de votre Compte AWS. Pour trouver le groupe de sécurité de ce VPC, sous Sécurité, choisissez Groupes de sécurité. Assurez-vous que la région `us-east-1` est sélectionnée.

Pour Linux, macOS ou Unix :

```
aws lambda create-function \  
--function-name AccessMemcached \  
--region us-east-1 \  
--zip-file fileb://my_deployment_package.zip \  
--role arn:aws:iam::123456789012:role/lambda-vpc-execution-role \  
--handler app.lambda_handler \  
--runtime python3.11 \  
--timeout 30 \  
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-security-group-id \  

```

Pour Windows :

```
aws lambda create-function ^  
--function-name AccessMemcached ^  
--region us-east-1 ^  
--zip-file fileb://path-to/my_deployment_package.zip ^  
--role arn:aws:iam::123456789012:role/lambda-vpc-execution-role ^  
--handler app.lambda_handler ^  
--runtime python3.11 ^  
--timeout 30 ^  
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-security-group-id ^  

```

Le cas échéant, vous pouvez charger le fichier .zip dans un compartiment Amazon S3 de la même région AWS, puis spécifier le nom du compartiment et de l'objet dans la commande précédente. Vous devez remplacer le paramètre `--zip-file` par le paramètre `--code`, comme illustré ci-après :

```
--code S3Bucket=bucket-name,S3Key=zip-file-object-key
```

Vous pouvez également créer la fonction Lambda à l'aide de la console AWS Lambda. Lors de la création de la fonction, choisissez un VPC pour la fonction Lambda, puis sélectionnez les groupes de sous-réseaux et de sécurité dans les champs fournis.

Étape suivante

[Étape 3 : Tester la fonction Lambda](#)

Étape 3 : Tester la fonction Lambda

Au cours de cette étape, vous allez appeler 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 cluster ElastiCache que vous avez spécifié dans votre code Lambda. La fonction Lambda récupère ensuite l'élément à partir du cache.

1. Appelez la fonction Lambda (AccessMemCache) à l'aide de la commande `invoke` AWS Lambda.

Pour Linux, macOS ou Unix :

```
aws lambda invoke \  
--function-name AccessMemCache \  
--region us-east-1 \  
output.txt
```

Pour Windows :

```
aws lambda invoke ^  
--function-name AccessMemCache ^  
--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 console [CloudWatch](#) et en choisissant le groupe de journaux de votre fonction (`/aws/lambda/AccessMemcached`). Le flux de journaux doit contenir un résultat similaire à ce qui suit :

```
Success: Inserted 05fcf2e4d6c942209acc89ea79b5b15e. Fetched  
05fcf2e4d6c942209acc89ea79b5b15e from Memcached.
```

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

Tutoriels et vidéos ElastiCache

Les didacticiels suivants traitent les tâches d'intérêt pour l'utilisateur Amazon ElastiCache.

- [Vidéos ElastiCache](#)
- [Didacticiel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un Amazon VPC](#)

Vidéos ElastiCache

Vous trouverez ci-après des vidéos pour vous aider à apprendre les concepts de base et avancés d'Amazon ElastiCache. Pour plus d'informations sur la formation AWS, veuillez consulter [Formation and Certification AWS](#).

Rubriques

- [Vidéos d'introduction](#)
- [Vidéos avancées](#)

Vidéos d'introduction

Les vidéos suivantes vous présentent Amazon ElastiCache.

Rubriques

- [AWS re:Invent 2020 : Nouveautés d'Amazon ElastiCache](#)
- [AWS re:Invent 2019 : Nouveautés d'Amazon ElastiCache](#)
- [AWS re:Invent 2017 : Nouveautés d'Amazon ElastiCache](#)
- [DAT204—Création d'applications évolutives sur les services NoSQL AWS \(re:Invent 2015\)](#)
- [DAT207—Optimisation des performances des applications avec Amazon ElastiCache \(AWS re:Invent 2013\)](#)

AWS re:Invent 2020 : Nouveautés d'Amazon ElastiCache

[AWS re:Invent 2020 : Nouveautés d'Amazon ElastiCache](#)

AWS re:Invent 2019 : Nouveautés d'Amazon ElastiCache

[AWS re:Invent 2019 : Nouveautés d'Amazon ElastiCache](#)

AWS re:Invent 2017 : Nouveautés d'Amazon ElastiCache

[AWS re:Invent 2017 : Nouveautés d'Amazon ElastiCache](#)

DAT204—Création d'applications évolutives sur les services NoSQL AWS (re:Invent 2015)

Dans cette session, nous évoquerons les avantages des bases de données NoSQL et nous ferons le tour des principaux services NoSQL offerts par AWS, Amazon DynamoDB et Amazon ElastiCache.

Puis, deux clients importants, Expedia et Mapbox, présenteront leurs cas d'utilisation et leurs contraintes d'architecture, et comment ils les ont traités grâce aux services NoSQL AWS, y compris leurs schémas techniques et leurs bonnes 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 NoSQL AWS \(re:Invent 2015\)](#)

DAT207—Optimisation des 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'améliorer la vitesse de réponse 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 bases 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 évoquerons plusieurs scénarios et étudierons des cas où l'utilisation de caches peut se révéler bénéfique, et discuterons des fonctionnalités d'Amazon ElastiCache.

[DAT207—Optimisation des performances des applications avec Amazon ElastiCache \(re:Invent 2013\)](#)

Vidéos avancées

Les vidéos suivantes traitent de rubriques plus avancées sur Amazon ElastiCache.

Rubriques

- [Concevoir pour réussir avec les meilleures pratiques Amazon ElastiCache \(re:Invent 2020\)](#)
- [Superchargez vos applications en temps réel avec Amazon ElastiCache \(re:Invent 2019\)](#)
- [Bonnes pratiques : migration de clusters Redis d'Amazon EC2 vers ElastiCache \(re:Invent 2019\)](#)
- [Mise à l'échelle d'une plateforme de sport fantastique avec Amazon ElastiCache et Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Redis 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 \(re:Invent 2016\)](#)
- [DAT407—Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402—Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)
- [DAT307 – Exploration de l'architecture et des modèles de conception Amazon ElastiCache \(re:Invent 2013\)](#)

[Concevoir pour réussir avec les meilleures pratiques Amazon ElastiCache \(re:Invent 2020\)](#)

Avec la croissance spectaculaire des applications critiques en temps réel basées sur Redis, la disponibilité, l'évolutivité et la sécurité sont devenues des facteurs déterminants. Découvrez les meilleures pratiques pour configurer Amazon ElastiCache pour réussir grâce à la mise à l'échelle en ligne, à la haute disponibilité dans les déploiements Multi-AZ et aux configurations de sécurité.

[Concevoir pour réussir avec les meilleures pratiques Amazon ElastiCache \(re:Invent 2020\)](#)

[Superchargez 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 booster les performances des applications en temps réel en utilisant Amazon ElastiCache entièrement géré et en mémoire pour des performances extrêmes, une haute évolutivité, une disponibilité et une sécurité élevées.

[Superchargez vos applications en temps réel avec Amazon ElastiCache \(re:Invent 2019:\)](#)

[Bonnes pratiques : migration de clusters Redis d'Amazon EC2 vers ElastiCache \(re:Invent 2019\)](#)

Gérer les clusters Redis par vous-même peut s'avérer complexe. Vous devez approvisionner le matériel, corriger les logiciels, sauvegarder les données et surveiller les charges de travail en permanence. Avec la nouvelle fonction de migration en ligne pour Amazon ElastiCache, vous pouvez désormais facilement déplacer vos données de Redis auto-hébergés sur Amazon EC2 vers Amazon ElastiCache entièrement géré, le mode cluster étant désactivé. Dans cette session, vous découvrirez le nouvel outil de migration en ligne, veuillez consulter une démo et, plus important encore, apprenez les bonnes pratiques pour une migration fluide vers Amazon ElastiCache.

[Bonnes pratiques : migration de clusters Redis d'Amazon EC2 vers ElastiCache \(re:Invent 2019\)](#)

Mise à l'échelle d'une plateforme de sport fantastique 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 cet exposé, Amit Sharma, directeur technique de Dream11, explique comment l'entreprise utilise Amazon Aurora et Amazon ElastiCache pour gérer le trafic flash, qui peut tripler dans une fenêtre de réponse de 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 : Mise à l'échelle d'une plateforme de sport fantastique avec Amazon ElastiCache et Amazon Aurora (STP11)

[Mise à l'échelle d'une plateforme de sport fantastique avec Amazon ElastiCache et Amazon Aurora STP11 \(re:Invent 2018\)](#)

Redis fiable et évolutif dans le cloud avec Amazon ElastiCache (re:Invent 2018)

Cette session couvre les fonctionnalités et les améliorations de notre service compatible avec REDIS, Amazon ElastiCache for Redis. Nous couvrons des fonctionnalités clés, telles que Redis 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 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)

Dans cette session, nous présentons un aperçu des coulisses pour en apprendre davantage sur la conception et l'architecture d'Amazon ElastiCache. Découvrez les points communs de nos offres Memcached et Redis, 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. Pendant cette vidéo, nous passerons en revue les bonnes pratiques, les modèles de conception et les anti-modèles d'ElastiCache.

[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 dessous d'Amazon ElastiCache pour en savoir plus sur sa structure et son architecture. Découvrez les points communs de nos offres Memcached et Redis, et comment les clients les ont utilisées pour leurs opérations de cache en mémoire afin de réduire la latence et d'améliorer le débit des applications. Pendant cette vidéo, nous passerons en revue les bonnes pratiques, les modèles de conception et les anti-modèles d'ElastiCache.

La vidéo présente les fonctionnalités suivantes°:

- ElastiCache for Redis
- Sécurité et chiffrement ElastiCache
- ElastiCache for Redis version 3.2.10

[DAT305—Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)

DAT306—Amazon ElastiCache Deep Dive (re:Invent 2016)

Découvrez les dessous d'Amazon ElastiCache pour en savoir plus sur sa structure et son architecture. Découvrez les points communs de nos offres Memcached et Redis, et comment les clients les ont utilisées pour leurs opérations de cache en mémoire afin de réduire la latence et d'améliorer le débit des applications. Au cours de cette session, nous passerons en revue les bonnes pratiques, les modèles de conception et les anti-modèles relatifs à ElastiCache.

[DAT306—Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)

DAT407—Amazon ElastiCache Deep Dive (re:Invent 2015)

Découvrez les dessous d'Amazon ElastiCache pour en savoir plus sur sa structure et son architecture. Observez les points communs entre les solutions Memcached et Redis et comment les clients les ont utilisées pour leurs opérations de cache en mémoire et ont obtenu de meilleurs délais de latence et de meilleures performances pour leurs applications. Au cours de cette session, nous passerons en revue les bonnes pratiques, les modèles de conception et les anti-modèles relatifs 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 habituels d'utilisation de cache, les moteurs Memcached et Redis, les modèles qui vous aideront à choisir le moteur le plus adapté à vos besoins, le hachage

cohérent, et tout ce qui vous permettra d'élaborer des applications rapides et scalables. Frank Wiebe, scientifique principal chez Adobe, explique en détail comment Adobe utilise Amazon ElastiCache pour améliorer l'expérience client et faire évoluer leur activité.

[DAT402—Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)

DAT307 – Exploration de l'architecture et des modèles de conception Amazon ElastiCache (re:Invent 2013)

Cette vidéo aborde la mise en cache, les stratégies de mise en cache, l'évolution et la surveillance. Nous comparons également les moteurs Memcached et Redis. Au cours de la session, nous passons également en revue les bonnes pratiques et les modèles relatifs à Amazon ElastiCache.

[DAT307 – Exploration de l'architecture et des modèles de conception Amazon ElastiCache \(AWS re:Invent 2013\)](#)

Choix des régions et des zones de disponibilité

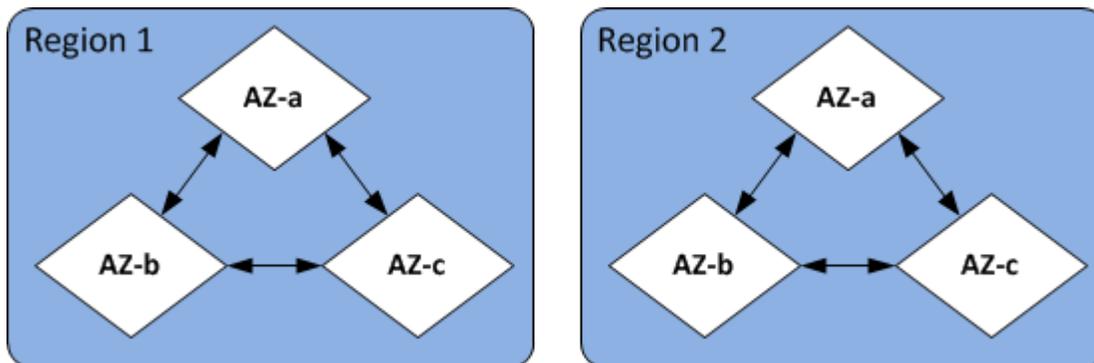
AWS Les ressources de cloud computing sont hébergées dans des centres de données à haute disponibilité. Pour offrir une évolutivité et une fiabilité supplémentaires, ces installations de centre de données sont situées dans différents emplacements physiques. Ces emplacements sont classés par régions et 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 points de terminaison pris en charge](#).



Régions et zones de disponibilité

Rubriques

- [Considérations sur les zones de disponibilité](#)
- [Régions et points de terminaison pris en charge](#)
- [Localisation de vos nœuds](#)

- [Utilisation de Local Zones avec ElastiCache](#)
- [Utilisation d'Outposts](#)

Considérations sur les zones de disponibilité

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 lors de la création de votre cache ElastiCache ou choisir des sous-réseaux automatiquement à partir de votre VPC par défaut.

Concevoir le vôtre ElastiCache pour le cluster 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 d'ID de nœud de cache répertoriés doit être égal au nombre de nœuds actuellement actifs moins la valeur de `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.

Régions et points de terminaison 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 dans différentes zones de disponibilité 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](#) au début de cette rubrique.

Régions dans lesquelles ElastiCache est pris en charge

Nom de région/Région	Point de terminaison	Protocole	
Région US East (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	

Nom de région/Région	Point de terminaison	Protocole	
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	
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 Asia Pacific (Seoul) ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS	

Nom de région/Région	Point de terminaison	Protocole	
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	
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	

Nom de région/Région	Point de terminaison	Protocole	
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	
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	

Nom de région/Région	Point de terminaison	Protocole	
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	
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	

Nom de région/Région	Point de terminaison	Protocole	
----------------------	----------------------	-----------	--

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, consultez [Types de nœuds pris en charge par région AWS](#).

Pour un tableau des AWS produits et services par région, voir [Produits et services par région](#).

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 (AZ). De plus, si vous choisissez de localiser vos nœuds dans plusieurs AZ (recommandé), cela vous ElastiCache permet de choisir l'AZ pour chaque nœud ou de les choisir ElastiCache pour vous.

En plaçant les nœuds dans différentes AZ, vous éliminez le risque qu'une défaillance, comme une panne de courant, dans une AZ mette en échec tout votre système.

Vous pouvez spécifier une AZ pour chaque nœud lors de la création d'un cluster ou lors de l'ajout de nœuds lorsque vous modifiez un cluster existant. Pour plus d'informations, consultez les ressources suivantes :

- [Création d'un cluster](#)
- [Modifier un ElastiCache cluster](#)
- [Ajout de nœuds à un cluster](#)

Utilisation de Local Zones avec ElastiCache

Une zone locale est une extension d'une région AWS qui est géographiquement proche de vos utilisateurs. Vous pouvez étendre n'importe quel Virtual Private Cloud (VPC) de la région AWS parente dans Local Zones en créant un sous-réseau et en l'attribuant 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.

Les zones locales (Local Zones) vous permettent de placer des ressources, par exemple un cluster ElastiCache, dans plusieurs emplacements proches de vos utilisateurs finaux.

Lorsque vous créez un cluster ElastiCache, 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 région AWS suivi d'un identifiant qui indique 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 limitations suivantes s'appliquent à ElastiCache pour Local Zones :

- 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 console Amazon EC2.

Pour de plus amples informations, veuillez consulter [Activation Local Zones](#) dans le Guide de l'utilisateur Amazon EC2 .

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 sous-réseaux ElastiCache dans la zone locale.

Lorsque vous créez un groupe de sous-réseaux ElastiCache, choisissez le groupe Zone de disponibilité pour la zone locale.

Pour de plus amples informations, veuillez consulter [Création d'un groupe de sous-réseaux](#) dans le Guide de l'utilisateur ElastiCache.

4. Créez un cluster ElastiCache for Memcached qui utilise le sous-réseau ElastiCache dans la zone locale.

Pour de plus amples informations, veuillez consulter [Création d'un cluster Memcached \(console\)](#).

Utilisation d'Outposts

AWS Outposts est un service entièrement géré qui étend l' AWS infrastructure, les services, les API et les outils aux locaux des clients. 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).
- 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 Memcached

1. Connectez-vous à la ElastiCache console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dans le volet de navigation, choisissez Memcached caches.
3. 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. Sous Memcached settings (Paramètres Memcached) :
 - a. Name (Nom) : saisissez un nom à donner au cluster Memcached.
 - b. Description : saisissez une description pour le cluster Memcached.
 - c. Compatibilité de la version du moteur : la version du moteur est basée sur la région Outpost AWS
 - d. Port : acceptez le port par défaut, 11211. Si vous préférez utiliser un autre port, saisissez le numéro de celui-ci.
 - 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. 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 nodes (Nombre de nœuds) : saisissez le nombre de nœuds que vous souhaitez dans votre cluster.
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.

- 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 ID de sous-réseau n'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 Memcached, sélectionnez un cluster appartenant à un AWS Outpost 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 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 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 \  
  \
```

Conception et gestion de votre propre cluster ElastiCache pour l'implémentation de Memcached

Si vous avez besoin de contrôler votre cluster ElastiCache avec précision, vous pouvez choisir de concevoir votre propre cluster. ElastiCache vous permet d'exploiter un cluster basé sur un nœud en choisissant le type de nœud, le nombre de nœuds et le placement des nœuds dans les zones de disponibilité AWS de votre cluster. Étant donné qu'ElastiCache est 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 [Configuration](#). Pour plus de détails sur la gestion, la mise à jour ou la suppression de nœuds ou de clusters, consultez [Gestion des nœuds](#). Pour obtenir une présentation des principaux composants d'un déploiement Amazon ElastiCache lorsque vous concevez votre propre cluster ElastiCache, consultez ces [concepts clés](#).

Rubriques

- [ElastiCache pour les composants et fonctionnalités de Memcached](#)
- [Gestion des clusters](#)
- [Gestion des nœuds](#)

ElastiCache pour les composants et fonctionnalités de Memcached

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

Rubriques

- [ElastiCache nœuds](#)
- [ElastiCache pour les clusters Memcached](#)
- [AWS Régions et zones de disponibilité](#)
- [ElastiCache pour les points de terminaison Memcached](#)
- [ElastiCache groupes de paramètres](#)
- [ElastiCache sécurité](#)
- [ElastiCache groupes de sous-réseaux](#)
- [ElastiCache pour les événements Memcached](#)

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 de Memcached. 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 [Mise à l'échelle ElastiCache pour Memcached](#).

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 pour fonctionner la plupart du temps et acheter des pay-as-you-go nœuds pour les

moments où vous devez parfois ajouter des nœuds. Pour plus d'informations sur les nœuds réservés, consultez [Nœuds réservés ElastiCache](#).

Le moteur Memcached prend en charge la découverte automatique. La découverte automatique est la possibilité pour les programmes client d'identifier automatiquement tous les nœuds d'un cluster de cache, et également d'initier et de maintenir des connexions à tous ces nœuds. Avec la découverte automatique, votre application n'a pas besoin de se connecter manuellement à des nœuds individuels. À la place, votre application se connecte à un point de terminaison de configuration. Le DNS du point de terminaison de la configuration contient les entrées CNAME pour chacun des points de terminaison des nœuds de cache. Ainsi, en se connectant au point de terminaison de configuration, votre application obtient immédiatement des informations sur tous les nœuds du cluster et peut se connecter à chacun d'entre eux. Vous n'avez pas besoin de coder en dur chaque point de terminaison d'un nœud de cache dans votre application. Pour plus d'informations, consultez [Découverte automatique](#).

Pour plus d'informations sur les nœuds, consultez [Gestion des nœuds](#).

ElastiCache pour les clusters Memcached

Un cluster memcached est un groupe logique d'un ou de plusieurs [ElastiCache nœuds](#). Les données sont partitionnées entre les nœuds d'un cluster memcached.

De nombreuses ElastiCache opérations ciblent les clusters :

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

Pour en savoir plus, consultez les rubriques connexes suivantes :

- [Gestion des clusters](#) et [Gestion des nœuds](#)

Informations sur les clusters, les nœuds et les opérations connexes.

- [AWS limites de service : Amazon ElastiCache](#)

Informations sur ElastiCache les limites, telles que le nombre maximum de nœuds ou de clusters.

Si vous devez dépasser ces limites, faites votre 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.

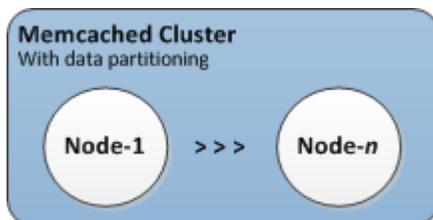
Configurations de clusters classiques

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 (AZ) 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 client ElastiCache pour un équilibrage de charge efficace](#). Dans le diagramme suivant, vous pouvez voir des exemples de clusters Memcached à nœud unique et à nœuds multiples.



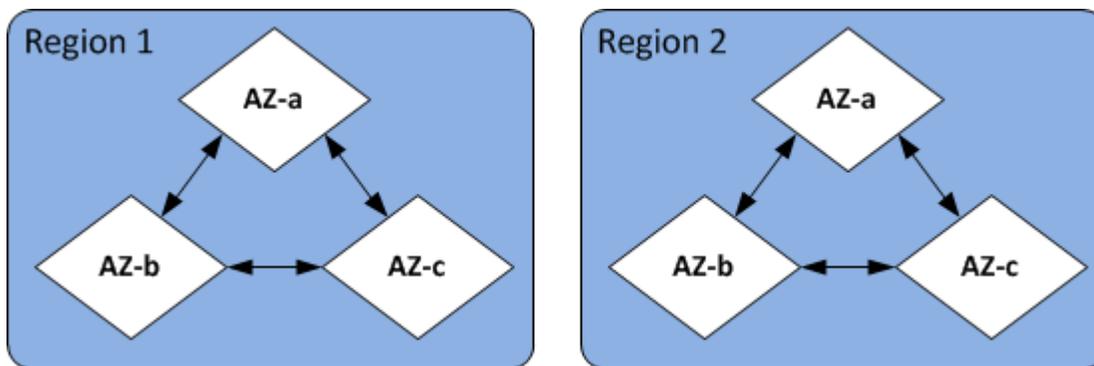
AWS Régions et zones de disponibilité

Amazon ElastiCache pour Memcached est disponible dans de nombreuses AWS régions du monde. Ainsi, vous pouvez lancer ElastiCache des clusters sur les sites qui répondent aux besoins de votre

entreprise. Par exemple, vous pouvez vous lancer dans la AWS région la plus proche de vos clients ou pour répondre à certaines exigences légales.

Par défaut, les AWS SDK AWS CLI, l' ElastiCache API et la ElastiCache console font référence à la région de l'ouest des États-Unis (Oregon). À mesure que la disponibilité ElastiCache s'étend à de nouvelles AWS régions, de nouveaux points de terminaison pour ces AWS régions peuvent également être utilisés dans vos requêtes HTTP AWS CLI, AWS les SDK et ElastiCache la console.

Chaque AWS région est conçue pour être complètement isolée des autres AWS régions. Chaque région contient plusieurs zones de disponibilité. En lançant vos nœuds dans différentes zones de disponibilité, vous pouvez obtenir la plus grande tolérance aux pannes possible. Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez [Choix des régions et des zones de disponibilité](#).



Pour plus d'informations sur AWS les régions prises en charge par ElastiCache et leurs points de terminaison, consultez [Régions et points de terminaison pris en charge](#).

ElastiCache pour les points de terminaison Memcached

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

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 plus d'informations, consultez [Points de terminaison](#) .

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](#) ([describe-engine-default-parameters](#)).

Pour des informations plus détaillées sur les groupes de ElastiCache paramètres, consultez [Configuration des paramètres de moteur à l'aide de groupes de paramètres](#).

ElastiCache sécurité

Pour renforcer la sécurité, l'accès aux ElastiCache nœuds est limité aux applications exécutées sur des instances Amazon EC2 figurant sur la liste blanche. Vous pouvez contrôler les instances Amazon EC2 qui peuvent accéder à votre cluster en utilisant des 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'instances Amazon EC2 s'exécutant sur des sous-réseaux spécifiques. Si vous choisissez d'exécuter votre cluster en dehors d'Amazon VPC, vous pouvez créer des groupes de sécurité pour autoriser les instances Amazon EC2 en cours d'exécution dans des groupes de sécurité Amazon EC2 spécifiques.

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 Amazon Virtual Private Cloud (Amazon VPC).

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 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 [Sécurité des VPC Amazon et ElastiCache](#), [Autoriser l'accès](#) et [Sous-réseaux et groupes de sous-réseaux](#).

ElastiCache pour les événements Memcached

Lorsque des événements importants se produisent sur un cluster de cache, ElastiCache envoie une notification à une rubrique Amazon SNS spécifique. Les événements importants peuvent inclure 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 clés, vous pouvez connaître l'état actuel de vos clusters et, en fonction de l'événement, prendre des mesures correctives.

Pour plus d'informations sur ElastiCache les événements, consultez [Surveillance des événements ElastiCache avec Amazon SNS](#).

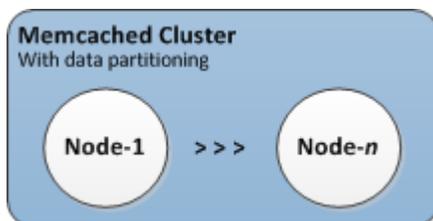
Gestion des clusters

Un cluster est une collection d'un ou plusieurs nœuds de cache, qui exécutent tous une instance du logiciel de moteur de cache pris en charge, Memcached ou . Lorsque vous créez un cluster, vous spécifiez le moteur et la version que tous les nœuds utiliseront.

Le graphique suivant illustre un cluster Memcached standard. Les clusters Memcached contiennent de 1 à 60 nœuds sur lesquels vous partitionnez vos données horizontalement.

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.

Un cluster Memcached type se présente comme suit.



La plupart des opérations ElastiCache sont effectuées au niveau du cluster. Vous pouvez définir un cluster avec un nombre spécifique de nœuds et un groupe de paramètres qui contrôle les propriétés de chaque nœud. Tous les nœuds au sein d'un cluster sont conçus pour avoir le même type de nœud et les mêmes paramètres et les mêmes configurations de 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 indique 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 à l'aide d'une instance Amazon EC2. 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 plus d'informations, consultez [Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC](#).

Pour obtenir la liste des versions de Memcached prises en charge, voir [Supportées ElastiCache pour les versions de Memcached](#).

Choix d'un type de réseau

ElastiCache prend en charge les versions 4 et 6 du protocole Internet (IPv4 et IPv6), ce qui vous permet de configurer votre cluster pour accepter :

- les connexions IPv4 uniquement,
- les connexions IPv6 uniquement,
- les connexions IPv4 et IPv6 (double pile)

IPv6 est pris en charge pour les charges de travail utilisant le moteur Memcached à partir de la version 1.6.6 sur toutes les instances créées sur le [système Nitro](#). Il n'y a aucun frais supplémentaire pour accéder à ElastiCache via IPv6.

Note

La migration des clusters créés avant la disponibilité de la connexion IPV6/double pile 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.

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 des adresses IP au sein de celui-ci à associer à vos nœuds. Les clusters ElastiCache nécessitent un sous-réseau à double pile auquel sont attribuées des adresses IPv4 et IPv6 pour fonctionner en mode double pile et un sous-réseau IPv6 uniquement pour fonctionner en IPv6 uniquement.

Utilisation de la double pile

Lorsque vous créez un cluster de cache et que vous choisissez la double pile comme type de réseau, vous devez ensuite désigner un type de découverte d'adresses IP, IPv4 ou IPv6. ElastiCache définit par défaut le type de réseau et la découverte d'adresses IP sur 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 maintenir la rétrocompatibilité avec tous les clients existants, la découverte d'adresses IP est introduite, ce qui vous permet de sélectionner le type d'adresse IP (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 grâce à la découverte automatique, car elle permet de migrer (ou de rétrograder) d'un type d'adresse IP de découverte IPv4 vers IPv6 sans interrompre le service.

Clusters ElastiCache à double pile prenant en charge TLS

Lorsque le protocole TLS est activé pour les clusters ElastiCache, les fonctions de découverte du cluster `config get cluster` renvoient des noms d'hôtes plutôt que des adresses IP. Les noms d'hôtes sont ensuite utilisés à la place des adresses IP pour se connecter au cluster ElastiCache et effectuer une liaison 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 de l'AWS Management Console

Lorsque vous créez un cluster de cache à l'aide de la AWS Management Console, sous Connectivity (Connectivité), choisissez un type de réseau, IPv4, IPv6 ou Dual Stack (Double pile). Si vous choisissez la double pile, vous devez sélectionner un Discovery IP type (Type d'adresse IP de découverte), IPv6 ou IPv4.

Pour de plus amples informations, veuillez consulter [Création d'un cluster Memcached \(console\)](#).

Utilisation de la CLI

Lorsque vous créez un cluster de cache à l'aide de l'interface de ligne de commande, vous utilisez la commande [create-cache-cluster](#) 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
```

Mise à niveau des données

Les clusters qui comprennent un groupe de réplication et qui utilisent un type de nœud de la famille r6gd ont leurs données hiérarchisées entre la mémoire et le stockage SSD local (disques SSD). La hiérarchisation des données fournit une option de rapport qualité-prix pour les charges de travail Redis en utilisant des disques SSD (Solid State Drive) moins coûteux dans chaque nœud de 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 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 Redis prises en charge dans ElastiCache. Vous n'avez besoin d'aucune modification côté client pour utiliser cette fonction.

Identifiez automatiquement les nœuds dans votre cluster

Pour les clusters qui utilisent le moteur Memcached, ElastiCache prend en charge la découverte automatique, la capacité pour les programmes clients d'identifier automatiquement tous les nœuds dans un cluster de cache et d'activer et de maintenir les connexions entre tous ces nœuds.

Note

La découverte automatique est ajoutée pour les clusters de cache s'exécutant sur Amazon ElastiCache Memcached.

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, la découverte automatique ne renverra que des adresses IPv4 ou IPv6, selon celle que vous sélectionnez. Pour plus d'informations, consultez [Choix d'un type de réseau](#).

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](#)
- [Comment fonctionne la découverte automatique](#)

- [Utilisation de la découverte automatique](#)
- [Connexion manuelle aux nœuds de cache](#)
- [Ajout de la découverte automatique à votre bibliothèque client](#)
- [Clients ElastiCache avec découverte automatique](#)

Avantages de la découverte automatique

Elle 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

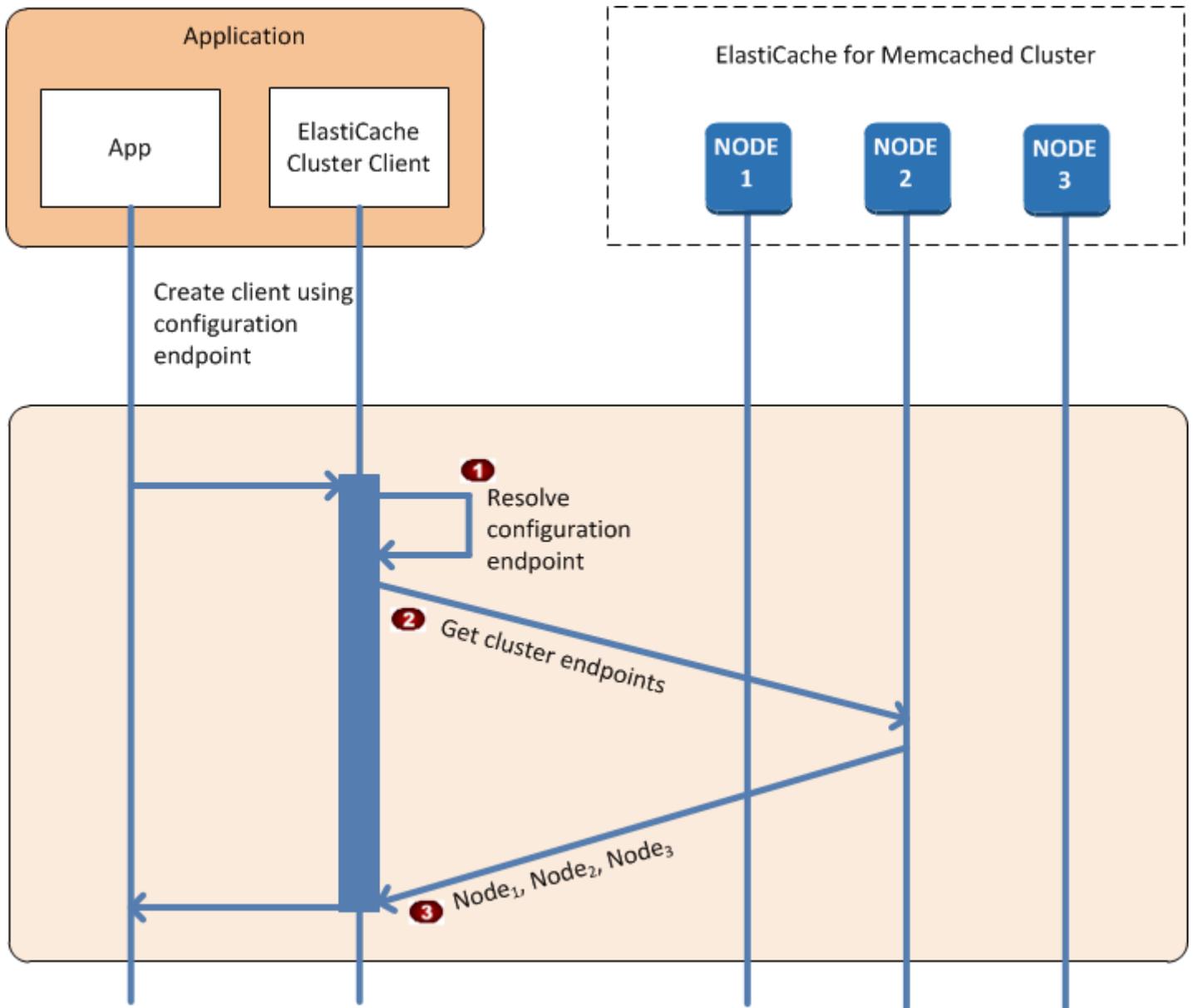
Rubriques

- [Connexion aux nœuds de cache](#)
- [Opérations de routine sur un cluster](#)
- [Autres opérations](#)

Cette section décrit comment les applications client utilisent le client de cluster ElastiCache pour gérer les connexions du nœud de cache et interagir avec les éléments de données dans le 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.

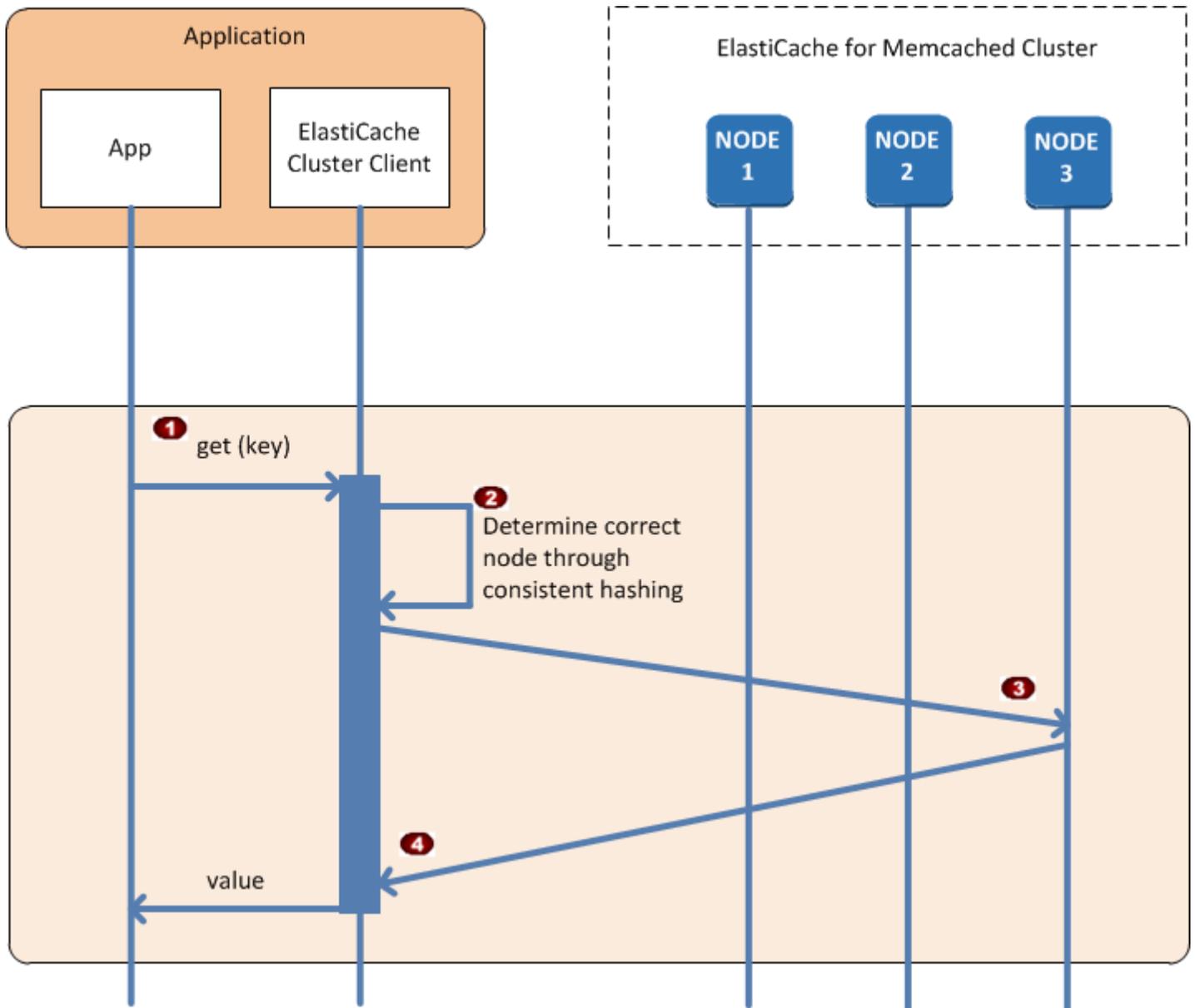
- Le client reçoit la liste actuelle des adresses IP et des noms d'hôtes des nœuds de cache. Il peut alors se connecter à tous les autres nœuds du cluster.

 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

Quand l'application s'est connectée à tous les nœuds de cache, le client de cluster ElastiCache détermine quels nœuds devront conserver chaque élément de données et dans quels nœuds ces éléments de données devront être recherchés par la suite. 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

En cas de défaillance d'un nœud, ElastiCache le supprime et active 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 la découverte automatique, suivez la procédure suivante :

- [Étape 1 : Obtenir le point de terminaison de configuration](#)
- [Étape 2 : Télécharger le client de cluster ElastiCache](#)
- [Étape 3 : Modifier votre programme d'application](#)

Étape 1 : 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 [Recherche des points de terminaison d'un cluster \(console\)](#)

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.

Exemple Recherche de points de terminaison à l'aide de la AWS CLI pour 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"
    }
]
}
```

Étape 2 : Télécharger le client de cluster ElastiCache

Pour pouvoir bénéficier de la découverte automatique, les programmes clients doivent utiliser le client de cluster ElastiCache. Le client de cluster ElastiCache 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 de cluster ElastiCache

1. Connectez-vous à la Console de gestion AWS et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Dans la console ElastiCache, choisissez ElastiCache Cluster Client (Client de cluster), puis Télécharger.

Le code source du client de cluster ElastiCache 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 cluster ElastiCache est publié sous la licence Amazon Software <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 cluster ElastiCache pour PHP, vous devez d'abord l'installer sur votre instance Amazon EC2. Pour de plus amples informations, veuillez consulter [Installation du client de cluster ElastiCache pour .NET](#).

Pour un client qui prend en charge TLS, téléchargez le binaire avec PHP version 7.4 ou supérieure.

Pour utiliser le client de cluster ElastiCache pour .NET, vous devez d'abord l'installer sur votre instance Amazon EC2. Pour de plus amples informations, veuillez consulter [Installation du client de cluster ElastiCache pour .NET](#).

Étape 3 : Modifier votre programme d'application

Modifiez votre programme d'application afin qu'il utilise la découverte automatique. Les sections suivantes expliquent comment utiliser le client de cluster ElastiCache 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 cluster ElastiCache pour Java

Le programme ci-dessous illustre comment utiliser le client de cluster ElastiCache pour connecter le point de terminaison de configuration d'un 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 cluster ElastiCache pour PHP

Le programme ci-dessous illustre comment utiliser le client de cluster ElastiCache pour connecter le point de terminaison de configuration d'un 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 cluster ElastiCache pour PHP, vous devez d'abord l'installer sur votre instance Amazon EC2. Pour de plus amples informations, veuillez consulter [Installation du client de cluster ElastiCache pour .NET](#).

```
<?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 obtenir un exemple d'utilisation du client de cluster ElastiCache en ayant activé TLS, consultez [Utilisation du chiffrement en transit avec PHP et Memcached](#).

Utilisation du client de cluster ElastiCache pour .NET

Note

Le client de cluster ElastiCache .NET est obsolète depuis mai 2022.

Le client .NET pour ElastiCache est disponible en 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 illustre l'utilisation du client de cluster ElastiCache pour se connecter à un point de terminaison de configuration d'un 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 ElastiCache 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

Si votre programme client n'utilise pas la découverte automatique, il peut se connecter manuellement à chaque nœud de cache. 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 commande AWS CLI `aws elasticache describe-cache-clusters` avec le paramètre `--show-cache-node-info`.

Exemple

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.

Ajout de la découverte automatique à votre bibliothèque client

Les informations de configuration pour la découverte automatique sont stockées de manière redondante dans chaque nœud du cluster de cache. 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.

Version du moteur de cache 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 binaires et ASCII Memcached par ElastiCache, et elle 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 voulez ajouter la prise en charge de la découverte automatique par l'intermédiaire du protocole binaire, reportez-vous au [code source du client de cluster ElastiCache](#).

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 <code>get</code> .	Oui
key	La clé sous laquelle la configuration du cluster est sauvegardée. Pour la découverte automatique, cette clé est appelée <code>cluster</code> .	Oui

Pour obtenir les informations de configuration du cluster, utilisez la commande suivante :

```
config get cluster
```

Version du moteur de cache inférieure à 1.4.14

Pour obtenir les informations de configuration du cluster, utilisez la commande suivante :

```
get AmazonElastiCache:cluster
```

Note

Ne modifiez pas la clé « `AmazonElastiCache:cluster` », car c'est là que résident les informations de configuration du cluster. Si vous remplacez cette clé, la configuration du client pourrait être incorrectement configurée pendant un court laps de temps (moins de 15 secondes) avant qu'ElastiCache corrige et mette à jour les informations de configuration automatiquement.

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.

Exemple

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.

Clients ElastiCache avec découverte automatique

Cette section décrit l'installation et la configuration des clients PHP et .NET d'ElastiCache.

Rubriques

- [Installation et compilation des clients de cluster](#)
- [Configuration des clients ElastiCache](#)

Installation et compilation des clients de cluster

Cette section couvre l'installation, la configuration et la compilation des clients de cluster de découverte automatique Amazon ElastiCache PHP et .NET.

Rubriques

- [Installation du client de cluster ElastiCache pour .NET](#)
- [Installation du client de cluster ElastiCache pour .NET](#)
- [Compilation du code source pour le client de cluster ElastiCache pour PHP](#)

Installation du client de cluster ElastiCache pour .NET

Note

Le client de cluster ElastiCache .NET est obsolète depuis mai 2022.

Vous pouvez trouver le code du client de cluster .NET ElastiCache 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 cluster ElastiCache sur les instances Amazon EC2. Pour plus d'informations sur la découverte automatique, consultez [Découverte automatique](#). Pour obtenir un exemple de code .NET pour utiliser le client, consultez [Découverte automatique avec DotNET](#).

Rubriques

- [Installation de .NET](#)
- [Téléchargement du client de cluster .NET ElastiCache pour ElastiCache](#)
- [Installation des assemblages AWS avec NuGet](#)

Installation de .NET

Vous devez avoir .NET 3.5 ou version ultérieure pour utiliser le kit AWS SDK pour .NET pour ElastiCache. Si vous ne disposez pas de .NET 3.5 ou version ultérieure, vous pouvez télécharger et installer la dernière version de <http://www.microsoft.com/net>.

Téléchargement du client de cluster .NET ElastiCache pour ElastiCache

Téléchargement du client de cluster .NET ElastiCache pour ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Dans le panneau de navigation, cliquez sur Client de cluster ElastiCache.
3. Dans la liste Download ElastiCache Memcached Cluster Client, sélectionnez .NET, puis cliquez sur Download.

Installation des assemblages AWS avec NuGet

NuGet est un système de gestion des packages pour la plateforme .NET. NuGet connaît des dépendances d'assemblage et installe tous les fichiers requis automatiquement. Les assemblages de NuGet sont stockés avec votre solution, plutôt que dans un emplacement central comme Program Files, de sorte que vous pouvez installer les versions spécifiques d'une application sans créer des problèmes de compatibilité.

Installation de NuGet

NuGet peut être installé à partir de la galerie d'installation sur MSDN ; accédez à <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Si vous utilisez Visual Studio 2010 ou une version ultérieure, NuGet est automatiquement installé.

Vous pouvez utiliser NuGet à partir de l'Explorateur de solutions ou de la console du Gestionnaire de package.

En utilisant NuGet à partir de l'Explorateur de solutions

Pour utiliser NuGet à partir de 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.

Pour utiliser NuGet à partir de l'Explorateur de solutions dans Visual Studio 2012 ou Visual Studio 2013

1. Dans le menu Tools, 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 qui est disponible via NuGet, comme les assemblages AWS SDK et AWS.Extensions, veuillez consulter le site Web de NuGet à l'adresse <http://www.nuget.org>. La page de chaque package comprend un exemple de ligne de commande pour installer le package à l'aide de la console et la liste des versions précédentes du package qui sont disponibles via NuGet.

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 cluster ElastiCache pour .NET

Cette section décrit comment installer, mettre à jour et supprimer les composants PHP pour le client de cluster ElastiCache sur les instances Amazon EC2. Pour plus d'informations sur la découverte automatique, consultez [Identifiez automatiquement les nœuds dans votre cluster](#). Pour un exemple de code PHP pour utiliser le client, consultez [Utilisation du client de cluster ElastiCache 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 d'utiliser la version adéquate du client de cluster ElastiCache pour PHP, vous devez savoir quelle version de PHP est installée sur votre instance Amazon EC2. Vous devez également savoir si votre instance Amazon EC2 exécute une version 64 bits ou 32 bits de Linux.

Pour déterminer la version PHP installée sur votre instance Amazon EC2

- 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 [Compilation du code source pour le client de cluster ElastiCache pour PHP](#).

Pour déterminer votre architecture AMI d'Amazon EC2 (64 bits ou 32 bits)

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
2. Dans la liste Instances, cliquez sur votre instance Amazon EC2.
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 cluster ElastiCache.

Pour télécharger le client de cluster ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Dans la console ElastiCache, choisissez ElastiCache Cluster Client (Client de cluster ElastiCache).
3. Dans la liste Download ElastiCache Memcached Cluster Client (Télécharger le client de cluster Memcached), choisissez le client de cluster ElastiCache correspondant à votre version PHP et à votre architecture AMI, puis choisissez le bouton Téléchargement.

Pour un client qui prend en charge TLS, téléchargez le binaire avec PHP version 7.4 ou supérieure.

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 extension ElastiCache php-memcached comme indiqué précédemment dans [Étapes d'installation pour les nouveaux utilisateurs](#).

Étapes d'installation pour les nouveaux utilisateurs

Rubriques

- [Installation de PHP 7.x – 8.x pour les nouveaux utilisateurs](#)
- [Installation de PHP 5.x pour les nouveaux utilisateurs](#)

Installation de PHP 7.x – 8.x pour les nouveaux utilisateurs

Rubriques

- [Pour installer PHP 7 – 8.x sur une AMI Amazon Linux 2](#)
- [Pour installer PHP 7 – 8.x sur une AMI Amazon Linux 201609](#)
- [Pour installer PHP 7.x – 8.x sur une AMI SUSE Linux 15](#)
- [Pour installer PHP 7.x – 8.x sur une AMI Ubuntu 22.04](#)

Pour installer PHP 7 – 8.x sur une AMI Amazon Linux 2

 Note

Si nécessaire, remplacez *PHP-7.x* avec la version que vous utilisez.

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez la commande suivante :

```
sudo yum install gcc-c++ zlib-devel
```

3. Installez PHP 7.x en utilisant `amazon-linux-extras`

Avec Amazon Linux 2, vous pouvez utiliser la bibliothèque Extras pour installer les mises à jour d'application et logicielles sur vos instances. Ces mises à jour logicielles sont appelées rubriques. Vous pouvez installer une version spécifique d'une rubrique ou omettre les informations de version pour utiliser la version la plus récente. Pour plus d'informations, veuillez consulter [Bibliothèque Extras \(Amazon Linux 2\)](#).

Pour cela, suivez les étapes suivantes :

- a. Tout d'abord, vérifiez si amazon-linux-extras est déjà installé.
- b. S'il n'est pas installé, utilisez la commande suivante pour l'installer :

```
sudo yum install -y amazon-linux-extras
```

- c. Vérifiez que la rubrique PHP 7.x est disponible sur la machine Amazon Linux 2 :

```
sudo amazon-linux-extras | grep php
```

- d. Dans la sortie, passez en revue tous les sujets de PHP 7 et sélectionnez la version que vous voulez :

```
sudo amazon-linux-extras enable php7.x
```

- e. Installez le package PHP à partir des référentiels. Par exemple :

```
sudo yum clean metadata
```

```
sudo yum install php php-devel
```

4. Téléchargez le client Amazon ElastiCache Cluster.

- Ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.

Sous le tableau de bord ElastiCache, accédez à Client de cluster ElastiCache, puis choisissez la version PHP7 que vous voulez.

- Dans la ligne de commande, remplacez PHP-7.X par la version de PHP souhaitée et remplacez ARCH par l'architecture souhaitée (X86 ou arm). Pour PHP >= 7.4, remplacez OpenSSL par la version d'OpenSSL souhaitée (openssl1.1 ou openssl3). Si vous utilisez PHP > 7.4, supprimez le suffixe OpenSSL.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.X/  
latest-64bit-<ARCH>-<OpenSSL>
```

- Utiliser `tar -zxvf` pour extraire le fichier téléchargé.

```
tar -zxvf latest-64bit-<ARCH>-<OpenSSL>
```

- Avec les autorisations racine, copiez le fichier artefact extrait `amazon-elasticache-cluster-client.so` dans `/usr/lib64/php/modules`.

```
sudo mv amazon-elasticache-cluster-client.so /usr/lib64/php/modules/
```

- Addition `extension=amazon-elasticache-cluster-client.so` dans le fichier `/etc/php.ini`
- Si vous avez téléchargé le client de cluster ElastiCache avec PHP 7.4 ou supérieur, installez OpenSSL 1.1.x ou supérieur. Instructions d'installation pour OpenSSL 1.1.1 :

```
sudo yum -y update
sudo yum install -y make gcc perl-core pcre-devel wget zlib-devel
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib64/libssl.so.1.1
```

Pour installer PHP 7 – 8.x sur une AMI Amazon Linux 201609

Note

Si nécessaire, remplacez *php7.x* avec la version que vous utilisez.

- Lancez une nouvelle instance depuis l'AMI. Pour plus d'informations, veuillez consulter [Étape 1 : Lancer une instance](#) dans le Guide de l'utilisateur Amazon EC2.
- Exécutez la commande suivante :

```
sudo yum install gcc-c++
```

3. Installez PHP

```
sudo yum install php7.x
```

4. Téléchargez le client Amazon ElastiCache Cluster.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/  
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.x/modules/.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.x/modules/
```

7. Créez le fichier 50-memcached.ini.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /  
etc/php-7.x.d/50-memcached.ini
```

8. Démarrez ou redémarrez votre serveur Apache.

```
sudo /etc/init.d/httpd start
```

Pour installer PHP 7.x – 8.x sur une AMI SUSE Linux 15

Note

Si nécessaire, remplacez *php7.x* avec la version que vous utilisez.

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez la commande suivante :

```
sudo zypper refresh
```

```
sudo zypper update -y
sudo zypper install gcc
```

3. Installez PHP

```
sudo yum install php7.x
```

or

```
sudo zypper addrepo //download.opensuse.org/repositories/devel:/languages:/php/
openSUSE_Leap_15.3/ php
```

4. Téléchargez le client de cluster Amazon ElastiCache, remplacez <ARCH> par l'architecture souhaitée (X86 ou arm). SUSE 15 est livré avec OpenSSL1.1 intégré, donc pour PHP >= 7.4 choisissez le binaire client avec OpenSSL1. Si vous utilisez PHP < 7.4, supprimez le suffixe OpenSSL.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/
latest-64bit-<ARCH>-openssl1.1
```

5. Extrayez latest-64bit.

```
tar -zxvf latest-64bit-<ARCH>-openssl1.1
```

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
```

Pour installer PHP 7.x – 8.x sur une AMI Ubuntu 22.04

Note

Si nécessaire, remplacez *php7.x* avec la version que vous utilisez.

1. Lancez une nouvelle instance depuis l'AMI.
2. Exécutez la commande suivante :

```
sudo apt-get update
sudo apt-get install gcc g++ make zlib1g zlib1g-dev
```

3. Installez PHP

- a. Instructions d'installation pour PHP 8.1 :

```
sudo apt install php8.1-cli php8.1-dev
```

- b. Instructions d'installation pour PHP 7.4 :

```
sudo apt -y install software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
sudo apt -y install php7.4
```

4. Téléchargez le client de cluster Amazon ElastiCache, remplacez <ARCH> par l'architecture souhaitée (X86 ou arm). Ubuntu 22.04 est livré avec OpenSSL3 intégré, donc pour PHP >= 7.4 choisissez le binaire client avec OpenSSL3. Si vous utilisez PHP < 7.4, supprimez le suffixe OpenSSL.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/
latest-64bit-<ARCH>-openssl3
```

5. Effectuez l'extraction de la dernière version 64 bits.

```
tar -zxvf latest-64bit-<ARCH>-openssl3
```

6. Avec l'autorisation racine, copiez le fichier artefact extrait `amazon-elasticache-cluster-client.so` dans le répertoire d'extension php `/usr/lib/php/20190902`. Si ce répertoire d'extension n'existe pas, vous pouvez le trouver en exécutant : `php -i | grep extension_dir`
7. Insérez la ligne `extension=amazon-elasticache-cluster-client.so` dans le fichier `/etc/php/7.x/cli/php.ini`.

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 package `php-memcached` adéquat pour votre instance Amazon EC2 et la version 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 la version actuelle :

```
$ sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

 Note

Veillez à utiliser la dernière version de l'artefact d'installation.

5. Avec l'autorisation racine/sudo, ajoutez un nouveau fichier nommé `memcached.ini` dans l'annuaire `/etc/php.d` 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 package `php-memcached` adéquat pour votre instance Amazon EC2 et la version 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 l'autorisation racine/sudo, ajoutez un nouveau fichier nommé `memcached.ini` dans l'annuaire `/etc/php.d` 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 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 package php-memcached adéquat pour votre instance Amazon EC2 et la version 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 l'autorisation racine/sudo, ajoutez un nouveau fichier nommé `memcached.ini` dans le répertoire `/etc/php5/cli/conf.d`, et insérez « `extension=<absolute path to amazon-ElastiCache-cluster-client.so>` » dans ce 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 package `php-memcached` adéquat pour votre instance Amazon EC2 et la version 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 l'autorisation racine/sudo, ajoutez un nouveau fichier nommé `memcached.ini` dans l'annuaire `/etc/php5/conf.d` et insérez `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 CentOS7 et Red Hat Enterprise Linux (RHEL) 7.1, `libsasl2.so.3` a remplacé `libsasl2.so.2`. Sur ces systèmes, lorsque vous chargez le client de cluster ElastiCache, il tente de trouver et de charger `libsasl2.so.2`, mais il n'y parvient pas. 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 ou supérieure](#)
- [Suppression d'une version antérieure de PHP 5](#)

Suppression d'une version antérieure de PHP 7 ou supérieure

Pour supprimer une version antérieure de PHP 7 ou supérieure

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.

Compilation du code source pour le client de cluster ElastiCache pour PHP

Cette section explique comment obtenir et compiler le code source du client de cluster ElastiCache pour PHP.

Il y a deux paquets que vous devez extraire de 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

Bibliothèques prérequis

- OpenSSL 1.1.0 ou supérieur (sauf si la prise en charge de TLS est désactivée par `./configure --disable-tls`).
- SASL (libsasl2, sauf si la prise en charge de SASL est désactivée par `./configure --disable-sasl`).

Pour compiler la bibliothèque aws-ElastiCache-cluster-client-libmemcached

1. Lancez une instance Amazon EC2.
2. Installez les dépendances de bibliothèque.
 - Sur l'AMI Amazon Linux 201509/l'AMI Amazon Linux 2

```
sudo yum -y update
sudo yum install gcc gcc-c++ autoconf libevent-devel make perl-core pcre-devel
wget zlib-devel
// Install OpenSSL 1.1.1
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib64/libssl.so.1.1
```

- Sur l'AMI Ubuntu 14.04 (non requise pour les versions d'Ubuntu fournies avec OpenSSL >= 1.1)

```
sudo apt-get update

sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev

// Install OpenSSL 1.1.1
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib/libssl.so.1.1 /usr/lib/x86_64-linux-gnu/libssl.so.1.1
```

3. Extrayez le référentiel et compilez le code.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached.git
cd aws-elasticache-cluster-client-libmemcached
touch configure.ac aclocal.m4 configure Makefile.am Makefile.in
mkdir BUILD
cd BUILD
../configure --prefix=<libmemcached-install-directory> --with-pic --disable-sasl
```

Si l'exécution de `../configure` ne parvient pas à trouver `libssl` (bibliothèque OpenSSL), il peut être nécessaire de modifier la variable d'environnement `PKG_CONFIG_PATH` :

```
PKG_CONFIG_PATH=/path/to/ssl/lib/pkgconfig ../configure --prefix=<libmemcached-install-directory> --with-pic --disable-sasl
```

Autrement, si vous n'utilisez pas TLS, vous pouvez le désactiver en exécutant :

```
make
sudo make install
../configure --prefix=<libmemcached-install-directory> --with-pic --disable-sasl --disable-tls
```

Compilation du client de découverte automatique ElastiCache Memcached pour PHP

Les sections suivantes décrivent comment compiler le client de découverte automatique Memcached ElastiCache

Rubriques

- [Compilation du client ElastiCache Memcached pour PHP 7 ou supérieur](#)
- [Compilation du client ElastiCache Memcached pour PHP 5](#)

Compilation du client ElastiCache Memcached pour PHP 7 ou supérieur

Remplacez PHP-7.x avec la version que vous utilisez.

Installez PHP :

```
sudo yum install -y amazon-linux-extras
sudo amazon-linux-extras enable php7.x
```

Exécutez l'ensemble de commandes dans le répertoire du code.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
phpize
mkdir BUILD
CD BUILD
../configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
```

Si l'exécution de `../configure` ne parvient pas à trouver `libssl` (bibliothèque OpenSSL), il peut être nécessaire d'ajuster la variable d'environnement `PKG_CONFIG_PATH` pour qu'elle pointe vers le répertoire du fichier `.PC` d'OpenSSL :

```
PKG_CONFIG_PATH=/path/to/ssl/lib/pkgconfig ../configure --with-libmemcached-dir=<path to libmemcached build directory> --disable-memcached-sasl
```

Autrement, si vous n'utilisez pas TLS, vous pouvez le désactiver en exécutant :

```
make
```

```
make install
../configure --with-libmemcached-dir=<path to libmemcached build directory> --disable-
memcached-sasl --disable-memcached-tls
```

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
```

Compilation du 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/aws-labs/aws-elasticache-cluster-client-memcached-for-
php/tree/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 clients ElastiCache

Un cluster ElastiCache est compatible avec les protocoles de Memcached. Le code, les applications et les outils les plus courants que vous utilisez déjà avec l'environnement Memcached existant fonctionnent de façon transparente avec le service.

Cette section contient des considérations spécifiques pour la connexion aux nœuds de cache dans ElastiCache.

Rubriques

- [Recherche des points de terminaison et des numéros de port des nœuds](#)
- [Connexion pour l'utilisation de la découverte automatique](#)
- [Noms DNS et réseau IP sous-jacent](#)

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 de gestion Amazon ElastiCache](#) et choisissez le moteur en cours d'exécution dans 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 (API ElastiCache)

Pour déterminer les points de terminaison et les numéros de port des nœuds de cache, utilisez l'action `DescribeCacheClusters` avec le paramètre `ShowCacheNodeInfo=true`.

Exemple

```
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 dans votre cluster](#).

Note

Pour l'instant, la découverte automatique n'est disponible que pour les clusters de cache sous Memcached.

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. Lorsque vous utilisez ElastiCache, l'API `DescribeCacheClusters` (ou l'utilitaire de ligne de commande `describe-cache-clusters`) 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.

Installations VPC

ElastiCache garantit que le nom DNS et l'adresse IP du nœud de cache restent identiques après récupération des nœuds de cache en cas de défaillance.

Installations non-VPC

ElastiCache garantit que le nom DNS d'un nœud de cache reste inchangé lorsque les nœuds de cache sont récupérés en cas de défaillance ; toutefois, l'adresse IP sous-jacente du nœud de cache peut varier.

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.

Préparation d'un cluster

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

Vous pouvez également créer un cluster ElastiCache à l'aide d'[AWS CloudFormation](#). Pour plus d'informations, consultez [AWS::ElastiCache::CacheCluster](#) dans le Guide de l'utilisateur AWS Cloud Formation, qui comprend des conseils sur la façon de mettre en œuvre cette approche.

Chaque fois que vous créez un cluster, 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](#)

- [Choix de la taille de votre nœud](#)

Déterminer les exigences

Préparation

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

Voulez-vous utiliser un service ElastiCache sans serveur ou basé sur une instance ?

Si vous souhaitez utiliser le cache sans serveur, assurez-vous simplement d'avoir correctement configuré votre VPC, vos sous-réseaux et vos groupes de sécurité. Pour en savoir plus, consultez [Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC](#). Si vous souhaitez utiliser le mode basé sur les instances ElastiCache, poursuivez votre lecture.

- 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 Memcached](#).

- 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 l'intérieur à AWS l'aide d'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 [Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC](#).

- 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 de paramètres](#).

- 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

- [Exigences relatives à la mémoire et au processeur](#)
- [Configuration du cluster Memcached](#)
- [Exigences relatives au dimensionnement](#)
- [Exigences relatives à l'accès](#)
- [Exigences relatives à la région, à la zone de disponibilité et à la zone locale](#)

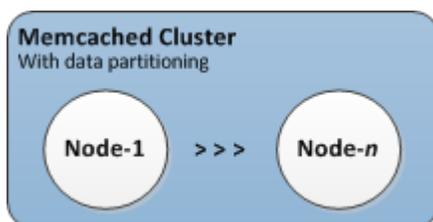
Exigences relatives à la mémoire et au processeur

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 [Identifiez automatiquement les nœuds dans votre cluster](#).



Pour de plus amples informations, veuillez consulter [Gestion des clusters](#).

Exigences relatives au dimensionnement

Tous les clusters peuvent être augmentés par la création d'un nouveau cluster doté d'un nouveau type de nœud plus grand. Lorsque vous mettez à l'échelle un cluster Memcached, le nouveau cluster démarre 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 [Mise à l'échelle ElastiCache pour Memcached](#) dans ce guide.

Exigences relatives à l'accès

De par leur conception, les ElastiCache clusters Amazon sont accessibles depuis des instances Amazon EC2. L'accès réseau à un ElastiCache cluster est limité au compte qui a créé le cluster. De ce fait, avant de pouvoir accéder à un cluster à partir d'une instance Amazon EC2, vous devez autoriser l'instance Amazon EC2 à accéder au cluster. Les étapes requises varient si vous avez lancé votre cluster dans un environnement EC2-VPC ou EC2-Classic.

Si vous avez lancé votre cluster dans un environnement EC2-VPC, vous devez accorder l'entrée 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 de sécurité ElastiCache. Pour plus d'informations, consultez [Accès à votre cluster](#) dans ce manuel.

Exigences relatives à la région, à la zone de disponibilité et à la zone locale

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 :

- [Régions et zones de disponibilité](#)
- [Zones locales](#)
- [Atténuation des défaillances](#)

Choix de la taille de votre nœud

La taille de nœud que vous sélectionnez pour votre cluster a un impact sur les coûts, les performances et la tolérance aux pannes.

Choix de la taille de votre 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, ElastiCache fournit un remplacement pour un nœud défaillant et il 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 \$
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, veuillez consulter la [Tarification Amazon ElastiCache](#).

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 une gestion simple de point de terminaison, vous pouvez utiliser la fonctionnalité de découverte automatique ElastiCache qui permet à des programmes client d'identifier automatiquement tous les nœuds dans un cluster. Pour de plus amples informations, veuillez consulter [Identifiez automatiquement les nœuds dans votre cluster](#).

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 CPU et le taux d'accès au cache à l'aide des métriques ElastiCache qui sont publiées sur Amazon CloudWatch. Pour plus d'informations sur les métriques CloudWatch pour ElastiCache, veuillez consulter [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#). 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

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 plus d'informations sur la découverte automatique, consultez [Identifiez automatiquement les nœuds dans votre cluster](#).

Pour créer un cluster Memcached, suivez les étapes décrites à l'[Étape 1 : Créer un cache](#).

Dès que l'état de votre cluster est disponible, vous pouvez accorder un accès Amazon EC2, vous y connecter et commencer à l'utiliser. Pour plus d'informations, consultez [Accès à votre cluster](#) et [Connexion manuelle aux nœuds de cache](#).

Important

Dès que votre cluster est disponible, vous êtes facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous ne l'utilisez pas activement. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter [Suppression d'un cluster](#).

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. veuillez consulter [Suppression d'un cluster](#).

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 (ElastiCache API)

Pour créer un cluster à l'aide de l' ElastiCache API, utilisez l'CreateCacheClusteraction.

Important

Dès que votre cluster est disponible, vous serez facturé pour chaque heure ou heure partielle où le cluster est actif, même si vous l'utilisez pas. Pour ne plus être facturé pour ce cluster, vous devez le supprimer. veuillez consulter [Suppression d'un cluster](#).

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>
```

Affichage des détails d'un cluster

Vous pouvez consulter les informations détaillées concernant un ou plusieurs clusters à l'aide de la console ElastiCache, de la AWS CLI ou de l'API ElastiCache.

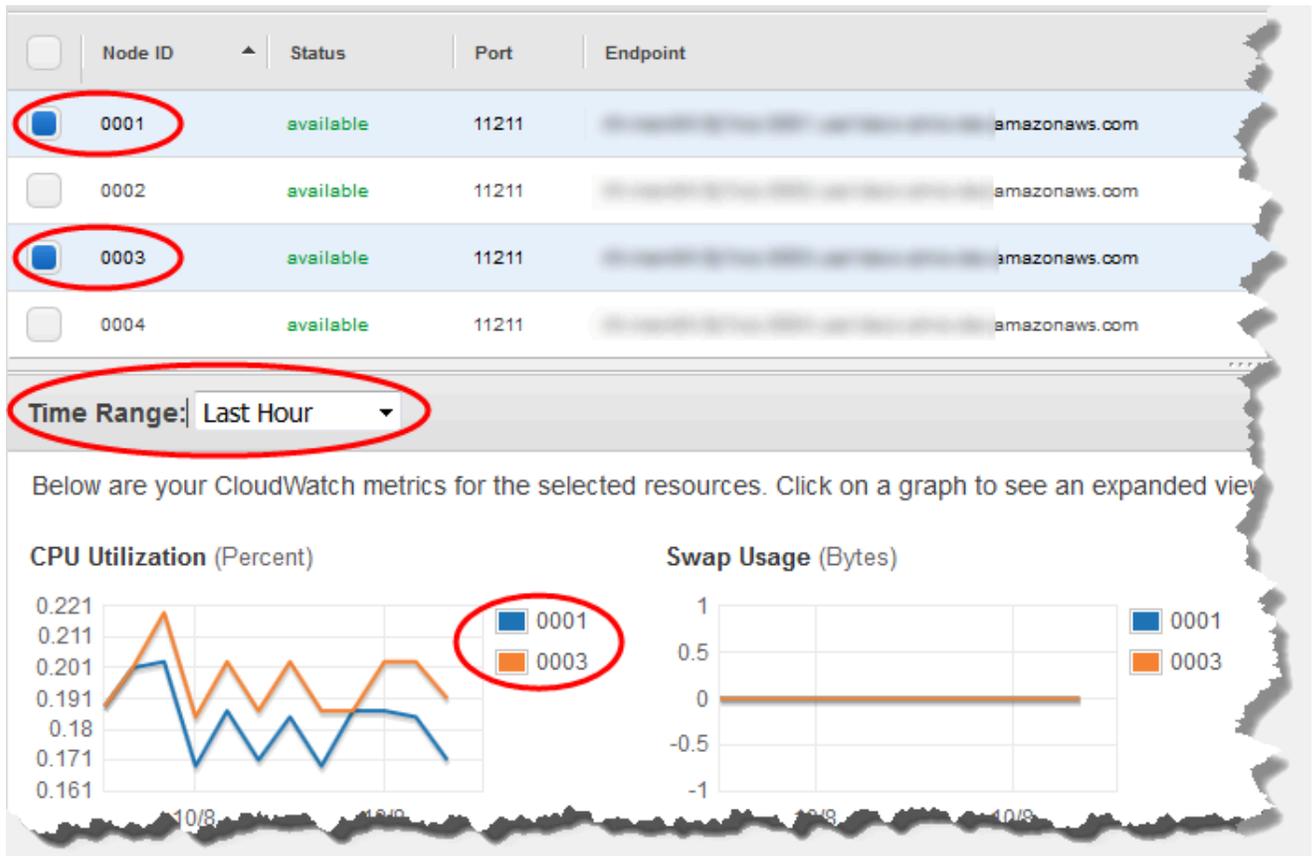
Affichage des détails d'un cluster (console)

Vous pouvez afficher les détails d'un cluster Memcached à l'aide de la console ElastiCache, la AWS CLI pour ElastiCache ou l'API ElastiCache.

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

Pour afficher les détails d'un cluster Memcached

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Depuis la liste située dans l'angle supérieur droit, choisissez la région AWS qui vous intéresse.
3. Dans le tableau de bord de la console ElastiCache, choisissez Memcached. Cela permet d'afficher une liste de tous vos clusters exécutant toutes les versions de Memcached.
4. Pour voir les détails d'un cluster, choisissez la zone à gauche du nom du cluster.
5. Pour afficher les informations de nœud :
 - a. Choisissez le nom du cluster.
 - b. Choisissez l'onglet Nodes.
 - c. Pour afficher les métriques d'un ou de plusieurs nœuds, choisissez la zone de gauche de l'ID de nœud, puis la plage de temps pour les métriques dans la liste Time range. Le choix de plusieurs nœuds génère des graphiques de superposition.



Métriques au cours de la dernière heure pour les deux nœuds Memcached

Affichage des détails d'un cluster (AWS CLI)

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

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

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Le code suivant affiche les détails de 25 clusters maximum.

```
aws elasticache describe-cache-clusters --max-items 25
```

Exemple

Pour Linux, macOS ou Unix :

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id my-cluster \  
  --show-cache-node-info
```

Pour Windows :

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id my-cluster ^  
  --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": "my-cluster.7ef-  
example.0001.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",  
          "CustomerAvailabilityZone": "us-west-2b"  
        },  
        {  
          "CacheNodeId": "0002",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "my-cluster.7ef-  
example.0002.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",
```

```
        "CustomerAvailabilityZone": "us-west-2b"
    },
    {
        "CacheNodeId": "0003",
        "Endpoint": {
            "Port": 11211,
            "Address": "my-cluster.7ef-
example.0003.usw2.cache.amazonaws.com"
        },
        "CacheNodeStatus": "available",
        "ParameterGroupStatus": "in-sync",
        "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",
        "CustomerAvailabilityZone": "us-west-2b"
    }
],
"CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "my-cluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
    "Port": 11211,
    "Address": "my-cluster.7ef-example.cfg.usw2.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 3,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
"SecurityGroups": [
    {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
    }
],
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
"PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
"CacheNodeType": "cache.m3.medium"
```

```
    }  
  ]  
}
```

Pour plus d'informations, veuillez consulter la rubrique AWS CLI pour ElastiCache [describe-cache-clusters](#).

Affichage des détails d'un cluster (API ElastiCache)

Vous pouvez afficher les détails d'un cluster à l'aide de l'action DescribeCacheClusters de l'API ElastiCache. 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, veuillez consulter la rubrique de référence de l'API ElastiCache [DescribeCacheClusters](#).

Modifier un ElastiCache cluster

En plus d'ajouter ou de supprimer des nœuds d'un cluster, vous aurez peut-être parfois besoin de faire des modifications sur un cluster existant, par exemple pour ajouter un groupe de sécurité, changer le créneau de maintenance ou un groupe de paramètres.

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

Lorsque vous 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 quand une modification de paramètre particulier sera appliquée, consultez Application des modifications dans les tableaux de [Paramètres spécifiques Memcached](#) et . Pour en savoir plus sur le redémarrage d'un cluster, consultez [Redémarrage d'un cluster](#).

À l'aide du 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/](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 :
 - Compatibilité de version de moteur
 - Groupes de sécurité VPC
 - Parameter Group

- Maintenance Window
- Rubrique pour la notification SNS

La case Apply Immediately (Appliquer immédiatement) concerne uniquement les modifications apportées à la version du moteur. 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.

7. Sélectionnez Modifier.

À l'aide du AWS CLI

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 mettre à niveau vers des versions de moteur plus récentes. Pour plus d'informations sur la procédure à utiliser, consultez [Versions du moteur et mise à niveau](#). 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 mettre à niveau vers des versions de moteur plus récentes. Pour plus d'informations sur la procédure à utiliser, consultez [Versions du moteur et mise à niveau](#). 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](#).

Redémarrage d'un cluster

Certaines modifications nécessitent le redémarrage du cluster pour qu'elles 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.

Lorsque vous redémarrez un cluster, 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 console ElastiCache, de la AWS CLI ou de l'API ElastiCache. Que vous utilisiez la console ElastiCache, la AWS CLI ou l'API ElastiCache, vous pouvez lancer un redémarrage uniquement sur un seul cluster. Pour redémarrer plusieurs clusters, vous devez répéter l'opération sur le processus ou l'action.

Utilisation de AWS Management Console

Vous pouvez redémarrer un cluster à l'aide de la console ElastiCache.

Pour redémarrer un cluster (console)

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Depuis la liste située dans l'angle supérieur droit, choisissez la région AWS 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 en choisissant le bouton à gauche du nom du cluster.

Choisissez Actions, puis Redémarrer.

Si vous choisissez plusieurs clusters, le bouton Redémarrer n'est pas actif.

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).

Utilisation de 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 plus d'informations, consultez [reboot-cache-cluster](#).

Utilisation de l'API ElastiCache

Pour redémarrer un cluster à l'aide de l'API ElastiCache, utilisez l'action `RebootCacheCluster`.

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 plus d'informations, consultez [RebootCacheCluster](#).

Ajout de nœuds à un 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 client ElastiCache pour un équilibrage de charge efficace](#).

Vous pouvez utiliser console de gestion ElastiCache, la AWS CLI ou l'API ElastiCache pour ajouter des nœuds dans votre cluster.

Utilisation de AWS Management Console

Rubriques

- [Pour ajouter des nœuds à un cluster \(console\)](#)

Pour ajouter des nœuds à un cluster (console)

La procédure suivante peut être utilisée pour ajouter des nœuds à un cluster.

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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.
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	<p>La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de suppression en attente.</p> <p>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.</p>
Scénario 2	Suppression	Création	<p>La nouvelle demande de création, en attente ou immédiate, remplace la demande de suppression en attente.</p> <p>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.</p>
Scénario 3	Création	Suppression	<p>La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de création en attente.</p> <p>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.</p>

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
Scénario 4	Création	Création	<p>La nouvelle demande de création est ajoutée à la demande de création en attente.</p> <p>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.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> Important</p><p>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.</p></div>

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.

Utilisation de AWS CLI

Pour ajouter des nœuds à un cluster à l'aide de l'AWS CLI, utilisez l'opération `modify-cache-cluster` de l'AWS CLI 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 attend le paramètre `cache-node-ids-to-remove` et une liste des nœuds à supprimer du cluster. Pour de plus amples informations, veuillez consulter [Utilisation de AWS CLI](#).
- `--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": "r1h-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",
}
}
```

Pour de plus amples informations, consultez la rubrique AWS CLI [modify-cache-cluster](#).

Utilisation de l'API ElastiCache

Pour ajouter des nœuds à un cluster (API ElastiCache)

- Appelez l'opération d'API `ModifyCacheCluster` avec les paramètres suivants :
 - `CacheClusterId` ID du cluster auquel vous souhaitez ajouter des nœuds.
 - `NumCacheNodes` Le paramètre `NumCacheNodes` 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 attend le paramètre

CacheNodeIdsToRemove avec une liste des nœuds à supprimer du cluster (voir [Utilisation de l'API ElastiCache](#)).

- ApplyImmediately spécifie s'il convient d'ajouter ces nœuds immédiatement ou lors du prochain créneau de maintenance.
- Region Spécifie la région AWS du cluster auquel vous souhaitez ajouter des nœuds.

L'exemple suivant montre un appel à ajouter des nœuds à un cluster.

Exemple

```
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, veuillez consulter la rubrique de référence de l'API ElastiCache [ModifyCacheCluster](#).

Suppression de nœuds d'un cluster

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 client ElastiCache pour un équilibrage de charge efficace](#).

Vous pouvez supprimer un nœud d'un cluster avec la AWS Management Console, la AWS CLI ou l'API ElastiCache.

Utilisation de AWS Management Console

Pour supprimer des nœuds d'un cluster (console)

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Depuis la liste située l'angle supérieur droit, choisissez la région AWS du cluster duquel 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 Delete node (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.

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	<p>La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de suppression en attente.</p> <p>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.</p>
Scénario 2	Suppression	Création	<p>La nouvelle demande de création, en attente ou immédiate, remplace la demande de suppression en attente.</p> <p>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.</p>
Scénario 3	Création	Suppression	<p>La nouvelle demande de suppression, en attente ou immédiate, remplace la demande de création en attente.</p> <p>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.</p>
Scénario 4	Création	Création	<p>La nouvelle demande de création est ajoutée à la demande de création en attente.</p>

Scénarios	Opérations en suspens	Nouvelle demande	Résultats
			<p>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.</p> <div data-bbox="727 527 1507 1031" style="border: 1px solid #f08080; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>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.</p> </div>

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.

Utilisation de AWS CLI

1. Identifiez les identifiants des nœuds que vous souhaitez supprimer. Pour de plus amples informations, veuillez consulter [Affichage des détails d'un cluster](#).
2. Utilisez l'opération `modify-cache-cluster` de la CLI avec une liste des nœuds à supprimer, comme dans l'exemple suivant.

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

- `--cache-cluster-id` ID du cluster de cache duquel vous voulez supprimer 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.

- `--cache-node-ids-to-remove` Une liste des ID de nœud que vous voulez 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 région AWS du cluster duquel vous voulez supprimer des nœuds.

L'exemple suivant supprime immédiatement le nœud 0001 du cluster my-cluster.

Pour Linux, macOS ou Unix :

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 2 \  
  --cache-node-ids-to-remove 0001 \  
  --region us-east-2 \  
  --apply-immediately
```

Pour Windows :

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 2 ^  
  --cache-node-ids-to-remove 0001 ^  
  --region us-east-2 ^  
  --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-east-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  

```

```
    "Address": "r1h-mem000.7ef-example.cfg.usw2.cache.amazonaws.com"
  },
  "CacheSecurityGroups": [],
  "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterStatus": "modifying",
  "NumCacheNodes": 3,
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
  "SecurityGroups": [
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "CacheSubnetGroupName": "default",
  "EngineVersion": "1.4.24",
  "PendingModifiedValues": {
    "NumCacheNodes": 2,
    "CacheNodeIdsToRemove": [
      "0001"
    ]
  },
  "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
  "CacheNodeType": "cache.m3.medium",
}
}
```

Pour de plus amples informations, veuillez consulter les rubriques AWS CLI [describe-cache-cluster](#) et [modify-cache-cluster](#).

Utilisation de l'API ElastiCache

Pour supprimer les nœuds à l'aide de l'API ElastiCache, appelez l'opération d'API `ModifyCacheCluster` avec l'ID de 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` La liste des ID de nœud 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 région AWS du cluster duquel vous voulez supprimer des nœuds.

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, veuillez consulter la rubrique de référence de l'API ElastiCache [ModifyCacheCluster](#).

Annulation des opérations d'ajout ou de suppression de nœud en attente

Si vous avez choisi de ne pas appliquer une modification immédiatement, l'opération a le statut pending jusqu'à ce qu'elle soit effectuée lors du prochain créneau de maintenance. Vous pouvez annuler toute opération en attente.

Pour annuler une opération en attente

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Depuis la liste située dans l'angle supérieur droit, choisissez la région AWS dans laquelle vous souhaitez annuler une opération en attente d'ajout ou de suppression.
3. Dans le volet de navigation, choisissez le moteur s'exécutant sur le cluster qui a des opérations en attente que vous souhaitez annuler. La liste des clusters exécutant le moteur choisi s'affiche.
4. Dans la liste des clusters, choisissez le nom du cluster, pas la case à gauche du nom du cluster, qui a des opérations en attente que vous souhaitez annuler.
5. 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.
6. Choisissez l'onglet Nodes.
7. Pour annuler toutes les opérations en attente, cliquez sur Cancel Pending. La boîte de dialogue Cancel Pending s'affiche.
8. Pour confirmer l'annulation de toutes les opérations en attente, choisissez le bouton En attente d'annulation ou, pour conserver ces opérations, choisissez Annuler.

Suppression d'un cluster

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

Utilisation de 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 AWS Management Console et ouvrez la console Amazon ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/>.
2. Dans le tableau de bord de la console ElastiCache, sélectionnez le moteur exécuté par le cluster à 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 dans la console ElastiCache. 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 de AWS CLI

Le code suivant supprime le cluster de cache `my-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, veuillez consulter la rubrique AWS CLI pour ElastiCache [delete-cache-cluster](#).

Utilisation de l'API ElastiCache

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, veuillez consulter la rubrique de référence de l'API ElastiCache [DeleteCacheCluster](#).

Accès à votre cluster

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

Si vous avez lancé votre instance ElastiCache dans un Amazon Virtual Private Cloud (Amazon VPC), vous pouvez accéder à votre instance ElastiCache depuis une instance Amazon EC2 dans le même Amazon VPC. En utilisant l'appairage de VPC, vous pouvez accéder à votre instance ElastiCache depuis une instance Amazon EC2 située dans un Amazon VPC différent.

Si vous avez lancé votre instance ElastiCache dans un environnement EC2 Classic, vous autorisez l'instance Amazon EC2 à accéder à votre cluster en donnant au groupe de sécurité Amazon EC2 associé à cette instance 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](#)

Octroi de l'accès à votre cluster

Vous avez lancé votre cluster dans EC2-VPC

Si vous avez lancé votre cluster dans un Amazon Virtual Private Cloud (Amazon VPC), vous pouvez vous connecter à votre cluster ElastiCache uniquement à partir d'une instance Amazon EC2 exécutée dans le même Amazon VPC. Dans ce cas, vous devez accorder l'accès au réseau au cluster.

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 AWS Management Console et ouvrez la console Amazon EC2 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 d'avoir créé un groupe de sécurité pour qu'il soit utilisé par ElastiCache, ce groupe de sécurité sera nommé default.
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. 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** .
 - e. Dans la zone Source, choisissez N'importe où qui a la plage de ports (0.0.0.0/0) afin que toute instance Amazon EC2 que vous lancez dans votre Amazon VPC puisse se connecter à vos nœuds ElastiCache.

Important

L'ouverture du cluster ElastiCache sur 0.0.0.0/0 n'expose pas le cluster sur Internet, car celui-ci n'a pas d'adresse IP publique et n'est donc pas accessible hors du VPC. Toutefois, le groupe de sécurité par défaut peut être appliqué aux autres instances Amazon EC2 dans le compte du client et ces instances peuvent voir une adresse IP publique. Si ces instances exécutent un service sur le port par défaut, ce service peut être exposé accidentellement. 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 Save (Enregistrer).

Lorsque vous lancez une instance Amazon EC2 dans votre Amazon VPC, cette instance sera en mesure de vous connecter à votre cluster ElastiCache.

Accès aux ressources ElastiCache depuis l'extérieur d'AWS

Amazon ElastiCache est un service AWS qui fournit un magasin de clés/valeurs en mémoire sur le Cloud. Le service est conçu pour être accessible exclusivement dans le cadre d'AWS. Toutefois, si le cluster ElastiCache est hébergé dans un VPC, vous pouvez utiliser une instance de traduction d'adresses réseau (NAT) pour fournir un accès extérieur.

Prérequis

Pour permettre l'accès à vos ressources ElastiCache en dehors d'AWS, les critères suivants doivent être satisfaits :

- 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 ressources ElastiCache en dehors d'AWS

La procédure suivante explique comment vous connecter à vos ressources ElastiCache 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*
- 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 ressources ElastiCache 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 NAT EC2 pour pouvoir accéder à ElastiCache en dehors d'AWS.

Pour plus d'informations sur la création d'une instance NAT, consultez [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
 - Une pour autoriser les connexions TCP à partir des clients de confiance pour chaque port de cache réacheminé depuis l'instance NAT (11211-11213).
 - Une deuxième pour autoriser un accès SSH aux clients de confiance.

Groupe de sécurité de l'instance NAT - Règles de trafic entrant

Type	Protocole	Plage de ports	Source
Destination	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

- Une règle de trafic sortant pour autoriser les connexions TCP vers le 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 de l'instance de cluster)

- Règle de trafic entrant 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-bd56b7da (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. 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
```

5. 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
```

6. 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 à partir d'une application exécutée dans le centre de données d'un client](#)
- [Instances NAT](#)
- [Configuration des clients ElastiCache](#)
- [Solution haute disponibilité des instances NAT sur Amazon VPC : un exemple](#)

Recherche de points de terminaison de connexion

Votre application se connecte à votre cluster à l'aide de points de terminaison. Un point de terminaison est l'adresse unique d'un nœud ou d'un cluster.

Quels points de terminaison utiliser

Pour le cache sans serveur ElastiCache avec Memcached, il vous suffit d'acquérir le DNS et le port du point de terminaison du cluster à partir de la console.

À partir d'AWS CLI, utilisez la commande `describe-serverless-caches` 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 une instance basée sur un cluster Memcached, si vous utilisez la découverte automatique, vous pouvez utiliser le point de terminaison de la 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.

Les sections suivantes vous guident à travers la découverte de points de terminaison dont vous aurez besoin pour le moteur utilisé.

Recherche des points de terminaison d'un cluster (console)

Tous les points de terminaison Memcached sont des points de terminaison de lecture/écriture. 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 « connaît » 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](#).

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)

Vous pouvez utiliser la AWS CLI pour Amazon ElastiCache afin de découvrir les points de terminaison des nœuds et clusters.

Rubriques

- [Recherche des points de terminaison des nœuds et des clusters \(AWS CLI\)](#)

Recherche des points de terminaison des nœuds et des clusters (AWS CLI)

Vous pouvez utiliser l'AWS CLI pour découvrir les points de terminaison d'un cluster et de ses nœuds avec la commande `describe-cache-clusters`. 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.

Exemple

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",
  }
]
}
```

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`.

Pour plus d'informations, consultez la rubrique [describe-cache-clusters](#).

Recherche des points de terminaison (API ElastiCache)

Vous pouvez utiliser l'API Amazon ElastiCache afin de découvrir les points de terminaison des nœuds et clusters.

Rubriques

- [Recherche des points de terminaison des nœuds et des clusters \(API ElastiCache\)](#)

Recherche des points de terminaison des nœuds et des clusters (API ElastiCache)

Vous pouvez utiliser l'API ElastiCache pour découvrir les points de terminaison d'un cluster et de ses nœuds avec l'action `DescribeCacheClusters`. 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.

Exemple

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`.

```
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`.

Gestion des nœuds

Un nœud est le plus petit élément constitutif d'un déploiement Amazon ElastiCache. 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 nœuds ElastiCache sont pris en charge, chacun ayant une quantité différente de mémoire et de puissance de calcul associée.

Pour une discussion plus détaillée sur la taille de nœud à utiliser, consultez [Choix de la taille de votre nœud Memcached](#).

Rubriques

- [Affichage de l'état du ElastiCache nœud](#)
- [Connexion aux nœuds](#)
- [Types de nœuds pris en charge](#)
- [Remplacement de nœuds](#)
- [Nœuds réservés ElastiCache](#)
- [Migration des nœuds de génération précédente](#)

Voici certaines opérations importantes impliquant des nœuds :

- [Ajout de nœuds à un cluster](#)
- [Suppression de nœuds d'un cluster](#)
- [Mise à l'échelle ElastiCache pour Memcached](#)
- [Recherche de points de terminaison de connexion](#)

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 une demande de modification du nœud d'un client.
updating	Facturé	<p>Un état de mise à jour indique qu'une ou plusieurs des conditions suivantes sont vraies pour le ElastiCache nœud Amazon :</p> <ul style="list-style-type: none">• Le ElastiCache nœud est en cours de correction 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 à jour pour le ElastiCache cluster.

Type	Facturé	Description
		<ul style="list-style-type: none">• 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 suppression du ElastiCache nœud est en attente.• Le mot de passe ElastiCache pour Redis est mis à jour/alterné à l'aide de. 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
<code>incompatible_parameters</code>	Non facturé	Amazon ElastiCache ne 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.
<code>incompatible_network</code>	Non facturé	Un état de réseau incompatible indique qu'une ou plusieurs des conditions suivantes sont vraies pour le nœud Amazon : ElastiCache <ul style="list-style-type: none">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	<p>Un état d'échec de restauration indique que l'une des conditions suivantes est vraie pour le nœud Amazon : ElastiCache</p> <ul style="list-style-type: none">• Les remplacements de nœuds ont échoué à plusieurs reprises en raison d'une capacité d'instance insuffisante. Cela se produit généralement lors de l'exécution de nœuds de génération précédente et qui le sont end-of-life. Toutefois, cela peut également se produire lors du remplacement de nœuds de la 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 suppression de ces nœuds, consultez Migration des nœuds de génération précédente.• La restauration de l'instantané RDB spécifié a échoué.• Le AWS compte du ElastiCache cluster a été suspendu.

Type	Facturé	Description
		<ul style="list-style-type: none">Le nœud est tombé en panne et n'a pas pu être restauré.
snapshotting	Facturé	ElastiCache crée un instantané du nœud ElastiCache for Redis.

Affichage de l'état du ElastiCache nœud avec la console

Pour afficher l'état d'un ElastiCache nœud 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, choisissez Redis 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.

Connexion aux nœuds

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 :

- [Recherche des points de terminaison d'un cluster \(console\)](#)
- [Recherche des points de terminaison \(AWS CLI\)](#)
- [Recherche des points de terminaison \(API ElastiCache\)](#)

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 instance Amazon EC2 à l'aide de l'utilitaire de connexion de votre choix.

Note

Pour obtenir des instructions sur la façon de se connecter à une instance Amazon EC2, consultez le [Guide de démarrage Amazon EC2](#).

2. Téléchargez et installez l'utilitaire telnet sur votre instance Amazon EC2. À l'invite de commande de votre instance Amazon EC2, entrez 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 instance Amazon EC2, entrez la commande suivante, en remplaçant le point de terminaison de votre nœud par celui indiqué 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

Pour plus d'informations sur la taille de nœud à utiliser, veuillez consulter [Choix de la taille de votre nœud Memcached](#).

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 de détails sur les performances pour chaque type de nœud, veuillez consulter [Types d'instances Amazon EC2](#).

- Usage général:
 - Génération actuelle :

Types de nœuds M6g (disponible uniquement pour le moteur Memcached à partir de la version 1.5.16).

cache.m6g.large, cache.m6g.xlarge, cache.m6g.2xlarge, cache.m6g.4xlarge, cache.m6g.8xlarge, cache.m6g.12xlarge, cache.m6g.16xlarge

Note

Pour connaître la disponibilité selon les régions, veuillez consulter [Types de nœuds pris en charge par région AWS](#).

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 M4 : cache.m4.large, cache.m4.xlarge, cache.m4.2xlarge, cache.m4.4xlarge, cache.m4.10xlarge

Types de nœuds T4g (disponible uniquement pour le moteur Memcached à partir de la version 1.5.16).

cache.t4g.micro, cache.t4g.small, cache.t4g.medium

Types de nœuds T3 : cache.t3.micro, cache.t3.small, cache.t3.medium

Types de nœuds T2: `cache.t2.micro`, `cache.t2.small`, `cache.t2.medium`

- Génération précédente : (non recommandé. Les clusters existants sont toujours pris en charge, mais la création de nouveaux clusters n'est pas prise en charge pour ces types.)

Types de nœuds T1: `cache.t1.micro`

Types de nœuds M1: `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`,
`cache.m1.xlarge`

Types de nœuds M3: `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`,
`cache.m3.2xlarge`

- Calcul optimisé :
 - Génération précédente : (non recommandée)

Types de nœuds C : `cache.c1.xlarge`

- Mémoire optimisée :
 - Génération actuelle :

(Les types de nœuds R6g ne sont disponibles uniquement qu'à partir de la version 1.5.16 du moteur Memcached.)

Types de nœud R6g : `cache.r6g.large`, `cache.r6g.xlarge`, `cache.r6g.2xlarge`,
`cache.r6g.4xlarge`, `cache.r6g.8xlarge`, `cache.r6g.12xlarge`,
`cache.r6g.16xlarge`

 Note

Pour connaître la disponibilité selon les régions, veuillez consulter [Types de nœuds pris en charge par région AWS](#).

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 R4: `cache.r4.large`, `cache.r4.xlarge`, `cache.r4.2xlarge`,
`cache.r4.4xlarge`, `cache.r4.8xlarge`, `cache.r4.16xlarge`

- Génération précédente : (non recommandée)

Types de nœuds M2: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

Types de nœuds R : `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`,
`cache.r3.4xlarge`, `cache.r3.8xlarge`

- Optimisés pour le réseau :
- Génération actuelle :

(Les types de nœuds C7gn ne sont disponibles uniquement qu'à partir de la version 1.6.6 du moteur Memcached.)

Types de nœuds C7gn : `cache.c7gn.large`, `cache.c7gn.xlarge`, `cache.c7gn.2xlarge`,
`cache.c7gn.4xlarge`, `cache.c7gn.8xlarge`, `cache.c7gn.12xlarge`,
`cache.c7gn.16xlarge`

Génération actuelle

Les tableaux suivants indiquent les bandes passante de base et extensibles pour les types d'instance qui utilisent le mécanisme de crédit d'E/S réseau pour dépasser le niveau de base de leur bande passante.

Note

Les types d'instances avec performances réseau extensibles utilisent un mécanisme de crédit d'E/S réseau pour dépasser leur bande passante de base dans la mesure du possible.

Général

Type d'instance	Version Memcached minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
<code>cache.m7g.large</code>		0,937	12,5
<code>cache.m7g.xlarge</code>		1,876	12,5
<code>cache.m7g.2xlarge</code>		3,75	15

Type d'instance	Version Memcached minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
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
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

Type d'instance	Version Memcached minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
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

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

Type d'instance	Version minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
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
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

Type d'instance	Version minimale prise en charge	Bande passante de référence (Gbit/s)	Bande passante de rafale (Gbit/s)
cache.r4.8xlarge	1.5,16	12	N/A
cache.r4.16xlarge	1.5,16	25	N/A

Optimisés pour le réseau

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

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 heurte pas à une baisse conséquente des performances lorsque le solde de crédits UC cumulés est bas. Pour plus d'informations, consultez [Crédits UC et performances de référence pour les instances à capacité extensible](#) dans le Guide de l'utilisateur Amazon EC2.

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 montre aussi le nombre maximal de crédits UC acquis qu'un nœud peut cumuler et le nombre de processeurs virtuels 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 *	vCPU	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
24	576	2	20 %	1,37	Jusqu'à 5 gigabits

Crédits UC gagnés par heure	Maximum de crédits gagnés pouvant être accumulés *	vCPU	Performances de base par vCPU	Mémoire (Gio)	Performances réseau
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 celles-ci, calculez l'utilisation de l'UC de référence pour le nœud en multipliant le pourcentage de processeurs virtuels par le nombre de processeurs virtuels.

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.

Informations connexes

- [Caractéristiques et détails des ElastiCache produits Amazon](#)
- [Paramètres propres au type de nœud Memcached](#)

Remplacement de nœuds

Amazon ElastiCache for Memcached met fréquemment à niveau sa flotte avec des correctifs et des mises à niveau appliquées aux instances de façon transparente. Toutefois, nous devons de temps en temps relancer vos nœuds ElastiCache for Memcached pour appliquer des mises à jour obligatoires du système d'exploitation de 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 entreprendre quand ElastiCache planifie le remplacement de l'un de vos nœuds Memcached.

- Ne rien faire – Si vous ne faites rien, ElastiCache remplace le nœud comme prévu. Quand ElastiCache remplace automatiquement le nœud par un nouveau nœud, le nouveau nœud est initialement vide.
- Changer votre fenêtre de maintenance – pour les événements de maintenance planifiés, vous recevez un e-mail ou une notification d'événement de la part d'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 [Modifier un ElastiCache cluster](#).

Note

La possibilité de modifier votre fenêtre de remplacement en déplaçant votre fenêtre de maintenance est disponible uniquement quand la notification ElastiCache 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](#).

- 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, consultez [Suppression de nœuds d'un cluster](#).
2. Ajouter un nouveau nœud au cluster. Pour obtenir des instructions, consultez [Ajout de nœuds à un 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 ElastiCache

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.

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 la tarification, veuillez consulter [Tarification Amazon ElastiCache](#).

Note

Les nœuds réservés ne sont pas flexibles ; ils s'appliquent exclusivement au type d'instance précis que vous réservez.

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 la tarification, veuillez consulter [Tarification Amazon ElastiCache](#).

La région AWS, le type de nœud et la durée du contrat doivent être choisis lors de l'achat et ne peuvent pas être modifiés ultérieurement.

Vous pouvez utiliser la AWS Management Console, la AWS CLI ou l'API ElastiCache pour répertorier et acheter les offres de nœuds réservés disponibles.

Pour plus d'informations sur les nœuds réservés, consultez [Nœuds réservés Amazon ElastiCache](#).

Rubriques

- [Offres relatives aux nœuds réservés standards](#)
- [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](#)

Offres relatives aux nœuds réservés standards

Lorsque vous achetez une instance de nœud réservé standard dans Amazon ElastiCache, vous achetez la garantie d'obtenir un tarif réduit sur un type d'instance de nœud et une région AWS spécifiques pour la durée de l'instance de nœud réservée. Pour utiliser une instance de nœud réservé d'Amazon ElastiCache, vous devez créer une nouvelle instance de nœud ElastiCache, tout comme vous le feriez pour une instance à la demande.

La nouvelle instance de nœud que vous créez doit correspondre exactement aux spécifications de l'instance de nœud réservé. Si les spécifications de la nouvelle instance de nœud coïncident avec une instance de nœud réservé existant dans votre compte, vous êtes facturé au tarif réduit correspondant à cette dernière. Dans le cas contraire, l'instance de nœud est facturée selon le tarif à la demande. Ces instances réservées standard sont disponibles à partir des familles d'instances R5 et M5.

Note

Les trois types d'offres présentés ci-dessous sont disponibles pour une période d'un an ou de trois ans.

Types d'offres

Sans frais initiaux L'instance réservée permet d'accéder à une instance ElastiCache réservée sans paiement initial. Pour votre instance ElastiCache réservée Sans frais initiaux, vous devez régler un tarif horaire avec remise pour chaque heure entrant dans le cadre de l'abonnement, indépendamment de son utilisation.

Frais initiaux partiels L'instance réservée exige qu'une partie de l'instance ElastiCache réservée soit payée d'avance. 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.

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 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 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 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 réservés pour utilisation faible, vous payez une cotisation unique accompagnée d'un taux d'utilisation horaire réduit lorsque votre nœud 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 utilisez 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	Medium	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)	Aucune	Tarif horaire le plus élevé. Appliqué lorsque le nœud est en cours d'exécution.	Coût horaire le plus élevé.

Pour plus d'informations, veuillez consulter [Tarification Amazon ElastiCache](#).

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 informations sur les offres de nœuds réservés disponibles et une tarification via l'AWS Management Console, l'AWS CLI et l'API ElastiCache.

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 des offres de nœuds réservés \(API ElastiCache\)](#)

Obtenir des informations sur des offres de nœuds réservés (console)

Pour obtenir une tarification et d'autres informations sur les offres de clusters réservés disponibles via l'AWS Management Console, procédez comme suit.

Pour obtenir des informations sur les offres de nœuds réservées disponibles

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 (Moteur), choisissez Memcached.
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 une tarification et d'autres informations sur les offres de clusters réservés disponibles, saisissez 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": "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, consultez [describe-reserved-cache-nodes-offerings](#) dans la référence AWS CLI.

Obtenir des informations sur des offres de nœuds réservés (API ElastiCache)

Pour obtenir une tarification et des informations sur les offres de nœuds réservés disponibles, appelez l'action `DescribeReservedCacheNodesOfferings`.

Exemple

```
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 de plus amples informations, veuillez consulter [DescribeReservedCacheNodesOfferings](#) dans la Référence de l'API ElastiCache.

Achat d'un nœud réservé

Les exemples suivants montrent comment acheter une offre de nœuds réservés via l'AWS Management Console, l'AWS CLI et l'API ElastiCache.

Important

Suivre les exemples de cette section entraîne des frais sur votre compte AWS 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 \(API ElastiCache\)](#)

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 l'AWS Management Console pour acheter l'offre de nœuds réservés via l'ID de l'offre.

Pour acheter des nœuds réservés

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 (Moteur), choisissez Memcached.
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, consultez [purchase-reserved-cache-nodes-offering](#) dans la référence AWS CLI.

Achat d'un nœud réservé (API ElastiCache)

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, veuillez consulter [PurchaseReservedCacheNodesOffering](#) dans la Référence de l'API ElastiCache.

Obtenir des informations sur vos nœuds réservés

Vous pouvez obtenir des informations sur les nœuds réservés que vous avez achetés via l'AWS Management Console, l'AWS CLI et l'API ElastiCache.

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 \(API ElastiCache\)](#)

Obtenir des informations sur vos nœuds réservés (console)

La procédure suivante utilise l'AWS Management Console pour acheter l'offre de nœuds réservés via l'ID de l'offre.

Pour obtenir des informations sur vos nœuds réservés achetés

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 pour votre type de compte AWS, 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",
```

```
"Duration": "31536000",
"ProductDescription": "memcached",
"OfferingType": "Medium Utilization",
"MaxRecords": 0
}
```

Pour plus d'informations, consultez [describe--reserved-cache-nodes](#) dans la référence AWS CLI.

Obtenir des informations sur vos nœuds réservés (API ElastiCache)

Pour obtenir des informations sur les nœuds réservés pour votre compte AWS, appelez l'opération `DescribeReservedCacheNodes`.

Exemple

```
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, veuillez consulter [DescribeReservedCacheNodes](#) dans la Référence de l'API ElastiCache.

Migration des nœuds de génération précédente

Les nœuds de génération précédente sont des types de nœuds qui sont en cours d'élimination progressive. Si vous n'avez aucun cluster existant utilisant 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, veuillez consulter [Types de nœuds pris en charge](#).

Migration de nœuds sur un cluster Memcached

Pour migrer ElastiCache for Memcached vers un autre type de nœud, vous devez créer un cluster, qui commence toujours par être vide et que votre application peut remplir.

Pour migrer votre type de nœud de cluster ElastiCache pour Memcached à l'aide de la console ElastiCache :

- 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 [Recherche des points de terminaison d'un cluster \(console\)](#)
- Supprimez l'ancien cluster. Pour de plus amples informations, veuillez consulter [Suppression d'un cluster](#).

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](#)
- [ElastiCache meilleures pratiques et stratégies de mise en cache](#)
- [Gestion de votre cluster auto-conçu](#)
- [Mise à l'échelle ElastiCache pour Memcached](#)
- [Étiquetage de vos ressources ElastiCache](#)
- [Utilisation du cadre Amazon ElastiCache Well-Architected](#)
- [Étapes de dépannage courantes et meilleures pratiques](#)
- [Étapes de dépannage supplémentaires](#)

Instantané et restauration

ElastiCache Les caches Amazon exécutant 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 Memcached Serverless 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](#)
- [Planification des sauvegardes automatiques](#)
- [Réalisation de sauvegardes manuelles](#)
- [Création d'une sauvegarde finale](#)
- [Description des sauvegardes](#)
- [Copie de sauvegardes](#)
- [Restauration à partir d'une sauvegarde dans un nouveau cache](#)
- [Suppression d'une sauvegarde](#)
- [Étiquetage des sauvegardes](#)

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 Redis ou Serverless Memcached.
- Au cours d'une période continue de 24 heures, vous ne pouvez pas créer plus de 24 sauvegardes manuelles par cache sans serveur.
- Pendant le processus de sauvegarde, vous ne pouvez exécuter aucune autre opération d'API ou de CLI sur le cache sans serveur.

Planification des sauvegardes automatiques

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 plus d'informations, consultez [Restauration à partir d'une sauvegarde dans un nouveau cache](#).

Lorsque vous planifiez des sauvegardes automatiques, vous devez à ElastiCache quel moment commencer à créer la sauvegarde. Vous pouvez définir le créneau de sauvegarde qui vous convient le mieux. Si vous ne spécifiez pas de fenêtre de sauvegarde, ElastiCache attribue-en une automatiquement.

Vous pouvez activer ou désactiver les sauvegardes automatiques lors de la création d'un cache Memcached Serverless à l'aide de la ElastiCache console, de l'API ou de l' AWS CLI API. ElastiCache Pour ce faire, cochez 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.

Création d'une sauvegarde manuelle (console)

Pour créer une sauvegarde d'un cache (console)

1. [Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Dans le volet de navigation, choisissez Memcached caches.
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
```

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 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 Memcached sans serveur 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

- [Utilisation de AWS Management 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 sans serveur](#)

Lors de la suppression d'un cache sans serveur

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
```

```
--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 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/](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-snapshots](#) dans le manuel de référence des commandes. AWS CLI

Copie de sauvegardes

Vous pouvez créer une copie de n'importe quelle sauvegarde, qu'elle ait été créée automatiquement ou manuellement.

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/](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.

Restauration à partir d'une sauvegarde dans un nouveau cache

Vous pouvez restaurer une sauvegarde existante dans un nouveau cache sans serveur .

Restauration d'une sauvegarde dans un cache sans serveur (console)

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/](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 cache sans serveur (AWS CLI)

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 :

Pour Windows :

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --snapshot-arns-to-restore Snapshot-ARN
```

Pour Windows :

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

```
--snapshot-arns-to-restore Snapshot-ARN
```

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/](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 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 plus d'informations, consultez [Ressources que vous pouvez étiqueter](#).

Les étiquettes de répartition des coûts sont un moyen de suivre vos coûts sur plusieurs AWS services en regroupant vos dépenses sur les factures par valeur de balise. Pour en savoir plus sur les balises de répartition des coûts, veuillez consulter [Utilisation des balises de répartition des coûts](#).

À l'aide de la 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 plus d'informations, voir [Surveillance des coûts avec des balises de répartition des coûts](#).

Versions du moteur et mise à niveau

Cette section présente les versions du moteur Memcached prises en charge et comment effectuer la mise à niveau.

Rubriques

- [ElastiCache pris en charge pour les versions Memcached](#)
- [Versions du moteur et mise à niveau](#)
- [Comment mettre à niveau les versions de moteur](#)

ElastiCache pris en charge pour les versions Memcached

ElastiCache prend en charge les versions Memcached ci-après ainsi que la mise à niveau vers les 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 version Memcached

- [Memcached version 1.6.22](#)
- [Memcached version 1.6.17](#)
- [Memcached version 1.6.12](#)
- [Memcached Version 1.6.6](#)
- [Memcached version 1.5.16](#)
- [Memcached Version 1.5.10](#)
- [Memcached Version 1.4.34](#)
- [Memcached Version 1.4.33](#)
- [Memcached Version 1.4.24](#)
- [Memcached Version 1.4.14](#)
- [Memcached Version 1.4.5](#)

Memcached version 1.6.22

ElastiCache for Memcached ajoute la prise en charge de 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](#) pour Memcached sur GitHub.

Memcached version 1.6.17

ElastiCache for Memcached ajoute la prise en charge de 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](#) de Memcached sur GitHub.

Memcached version 1.6.12

ElastiCache for Memcached ajoute la prise en charge de Memcached version 1.6.12 et le 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](#) de Memcached sur GitHub.

Memcached Version 1.6.6

ElastiCache for Memcached ajoute la prise en charge de Memcached version 1.6.6. Sans aucune nouvelle fonctionnalité, il inclut des correctifs de bogue et des mises à jour cumulatives à partir des versions [Memcached 1.5.16](#). ElastiCache for Memcached n'inclut pas la prise en charge de [Extstore](#).

Pour plus d'informations, veuillez consulter les [notes de mise à jour 166](#) de Memcached sur GitHub.

Memcached version 1.5.16

ElastiCache for Memcached ajoute la prise en charge de Memcached version 1.5.16. 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, veuillez consulter les [notes de mise à jour de Memcached version 1.5.16](#) sur GitHub.

Memcached Version 1.5.10

ElastiCache for Memcached version 1.5.10 prend en charge les fonctionnalités Memcached suivantes :

- 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 plus d'informations, consultez [Modifications des paramètres Memcached 1.5.10](#).

Les améliorations de Memcached ajoutées depuis ElastiCache pour Memcached version 1.4.34 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 GitHub.

Memcached Version 1.4.34

ElastiCache for Memcached version 1.4.34 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 GitHub.

Memcached Version 1.4.33

Les améliorations Memcached 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](#).

Memcached Version 1.4.24

Les améliorations Memcached 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.

Memcached Version 1.4.14

Les améliorations Memcached ajoutées 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.

Memcached Version 1.4.5

Memcached version 1.4.5 a été le moteur initial et la version prise en charge par Amazon ElastiCache for Memcached.

Versions du moteur et mise à niveau

Les versions MAJOR concernent les modifications incompatibles avec l'API, tandis que les versions MINOR concernent les nouvelles fonctionnalités ajoutées de manière rétrocompatible. Les versions PATCH sont destinées aux correctifs de bogues rétrocompatibles et aux modifications non fonctionnelles.

Gestion des versions pour ElastiCache sans serveur

ElastiCache sans serveur applique automatiquement les dernières versions logicielles MINOR et PATCH à votre cache, sans aucun impact ni aucune interruption de service pour votre application. Aucune action de votre part n'est nécessaire.

Lorsqu'une nouvelle version MAJOR est disponible, ElastiCache sans serveur 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.

Gestion des versions pour les clusters ElastiCache auto-conçus

Lorsque vous utilisez des clusters ElastiCache auto-conçus, vous pouvez contrôler à quel moment mettre à niveau le logiciel qui alimente votre cluster de cache vers les nouvelles versions prises en charge par ElastiCache. Vous pouvez contrôler à quel moment mettre à niveau votre cache vers les dernières versions MAJOR, MINOR et PATCH 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.

Pour mettre à niveau vers une version plus récente de Memcached, modifiez le cluster de cache en indiquant 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 plus d'informations, consultez [Modification des clusters](#).

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](#).

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. Elles ne s'appliquent pas à ElastiCache sans serveur.

Lors de la mise à niveau d'un cluster auto-conçu, prenez en compte les éléments 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. Cependant, ElastiCache se réserve le droit de corriger votre cluster en votre nom dans l'éventualité improbable d'une faille de vulnérabilité critique dans le système ou le logiciel du 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.

Comment mettre à niveau les versions de moteur

Pour démarrer les mises à niveau de la version de votre cluster, vous le modifiez et indiquez une nouvelle version du moteur. Pour ce faire, utilisez la console ElastiCache, la AWS CLI, ou l'API ElastiCache :

- Pour utiliser la AWS Management Console, consultez [Modification des clusters via la console](#).
- Pour utiliser AWS CLI, consultez [Modification des clusters avec l'interface de ligne de commande](#).
- Pour utiliser l'API ElastiCache, consultez [Modification des clusters via l'API](#).

Comment mettre à niveau les versions de moteur

Pour démarrer les mises à niveau de la version de votre cluster, vous le modifiez et indiquez une nouvelle version du moteur. Pour ce faire, utilisez la console ElastiCache, la AWS CLI, ou l'API ElastiCache :

- Pour utiliser l'AWS Management Console, consultez – [À l'aide du AWS Management Console](#).
- Pour utiliser l'AWS CLI, veuillez consulter [À l'aide du AWS CLI](#).
- Pour utiliser l'API ElastiCache, veuillez consulter [Utilisation de l' ElastiCache API](#).

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 avec les clients Memcached](#)
- [Commandes Memcached prises en charge](#)
- [Stratégies de mise en cache](#)

Bonnes pratiques avec les clients Memcached

Pour en savoir plus sur les bonnes pratiques en matière d'interaction avec les ressources ElastiCache en utilisant les bibliothèques client Memcached open source couramment utilisées, consultez les rubriques ci-dessous.

Rubriques

- [Configuration de votre client ElastiCache pour un équilibrage de charge efficace](#)
- [Exemples de client IPv6](#)

Configuration de votre client ElastiCache pour un équilibrage de charge efficace

Note

Cette section s'applique aux clusters Memcached à plusieurs nœuds auto-conçus.

Pour travailler efficacement avec plusieurs nœuds Memcached ElastiCache, vous devez être en mesure de répartir vos clés de cache entre les nœuds. Un moyen simple pour équilibrer un cluster avec n nœuds consiste à calculer le hachage de la clé de l'objet et à appliquer l'opérateur mod au résultat, soit $n - \text{hash}(\text{key}) \bmod n$. 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 utilisez Java, PHP ou .NET, nous vous conseillons d'utiliser l'une des bibliothèques client Amazon ElastiCache.

Hachage cohérent utilisant la technologie Java

Le client Java Memcached ElastiCache est basé sur le client Java open source spymemcached, qui a des capacités de hachage cohérent intégrées. La bibliothèque comprend une classe

KetamaConnectionFactory qui met en œuvre le hachage cohérent. Par défaut, le hachage cohérent est désactivé dans spymemcached.

Pour de plus amples informations, consultez la documentation KetamaConnectionFactory à l'adresse [KetaConnectionFactory](#).

Hachage cohérent utilisant la technologie PHP

Le client PHP Memcached ElastiCache 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 sur <http://php.net/manual/en/memcached.configuration.php>. Notez spécifiquement le paramètre `memcached.sess_consistent_hash`.

Hachage cohérent utilisant la technologie .NET

Le client .NET Memcached ElastiCache 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 documentation `memcached/locator` sur <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator>.

Exemples de client IPv6

Note

Cette section s'applique aux clusters Memcached auto-conçus.

ElastiCache est compatible avec le logiciel libre Memcached. Cela signifie que les clients open source pour Memcached qui prennent en charge les connexions IPv6 devraient être en mesure de se connecter à IPv6 activé ElastiCache pour les clusters Memcached. En outre, les clients suivants ont

été spécifiquement validés pour fonctionner avec toutes les configurations de type de réseau prises en charge :

Vous trouverez ci-dessous les meilleures pratiques pour interagir avec les ElastiCache ressources compatibles IPv6 avec les bibliothèques clientes open source couramment utilisées. Vous pouvez consulter [les meilleures pratiques existantes en matière d'interaction pour ElastiCache obtenir des](#) recommandations sur la configuration des clients pour les ElastiCache ressources. Toutefois, certaines précautions méritent d'être prises en compte lors de l'interaction avec des ressources prenant en charge IPv6.

Clients validés

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

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 de découverte d'adresses IP 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 les adresses IP de tous les nœuds du cluster qui utilisent le protocole mis à jour, la propagation des modifications est terminée.

Exemple Java :

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4
```

```
Set<String> nodes;

do {
    nodes =
    client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toSet());

    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 le protocole TLS est activé pour les ElastiCache clusters, les fonctions de découverte des clusters renvoient des noms d'hôtes plutôt que des adresses IP. Les noms d'hôtes sont ensuite utilisés à la place des adresses IP pour se connecter au ElastiCache cluster et effectuer une prise de contact 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 prenant en charge à la fois IPv4 et IPv6, Java préfère par défaut IPv4 à IPv6 pour des raisons de 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, `-Djava.net.preferIPv6Stack=true`. Le paramètre `-Djava.net.preferIPv4Stack=true` signifie que la JVM n'établira plus de connexions IPv6.

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 préfèrent IPv6 à 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 afin de préférer IPv4 lors de la connexion à un ElastiCache cluster, mettez à jour la priorité de la plage d'adresses CIDR comprenant les adresses IP du cluster afin qu'elle soit supérieure à la priorité des connexions IPv6 par défaut. Les connexions IPv6 ont par

défaut une priorité de 40. Par exemple, en supposant que le cluster se trouve dans un sous-réseau avec l'adresse CIDR 172.31.0.0/16, la configuration ci-dessous inciterait les clients à préférer les connexions IPv4 à ce cluster.

```
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 sur `gai.conf` sont disponibles sur la [Linux main page](#) (Page principale de 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 les connexions IPv4 aux connexions IPv6 pour la plage CIDR spécifiée. Par exemple, en supposant que le cluster se trouve dans un sous-réseau avec l'adresse CIDR 172.31.0.0:0/16 exécutant `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15`, le tableau de priorité suivant inciterait les clients à préférer IPv4 lors de la connexion au cluster.

```
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
```

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

Stratégies de mise en cache

Dans la rubrique suivante, vous trouverez des stratégies pour remplir et maintenir votre cache.

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

- [Chargement différé](#)
- [Écriture simultanée](#)
- [Ajout d'une durée de vie](#)
- [Rubriques en relation](#)

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 qui se trouve entre votre application et le magasin de données (base de données) auquel il accède. Chaque fois que votre application demande des données, elle effectue tout d'abord la requête dans la mémoire cache d'ElastiCache. Si les données existent dans le cache et sont à jour, ElastiCache renvoie les données à 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.

Time to live (TTL) est une valeur entière qui spécifie le nombre de secondes qui doivent s'écouler jusqu'à l'expiration de la clé. 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 la [commande Memcached set](#).

Exemples de pseudocode TTL

Le code suivant est un exemple de pseudo-code de logique d'écriture simultanée comportant 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](#)
- [Mise à l'échelle ElastiCache pour Memcached](#)

Gestion de votre cluster auto-conçu

Cette section contient des rubriques qui vous aident à gérer vos clusters auto-conçus.

Note

Ces rubriques ne s'appliquent pas à ElastiCache sans serveur.

Rubriques

- [Gestion de la maintenance](#)
- [Configuration des paramètres de moteur à l'aide de groupes de paramètres](#)

Gestion de la maintenance

Chaque cluster a un créneau de maintenance hebdomadaire au cours duquel toutes les modifications systèmes seront appliquées. Si vous ne spécifiez pas de fenêtre de maintenance préférée lorsque vous créez ou modifiez un cluster, ElastiCache assigne une fenêtre de maintenance de 60 minutes dans le créneau de maintenance de votre région, un jour de la semaine choisi au hasard.

Ce créneau de maintenance de 60 minutes est choisi de manière aléatoire sur un bloc horaire de 8 heures par région. Le tableau suivant répertorie pour les différentes régions les blocs de temps à partir desquels les créneaux de maintenance par défaut sont alloués. Vous pouvez choisir un créneau de maintenance préféré en dehors du créneau de maintenance de votre région.

Code région	Nom de la région	Fenêtre de maintenance régionale
ap-northeast-1	Région Asie-Pacifique (Tokyo)	13:00–21:00 UTC
ap-northeast-2	Région Asia Pacific (Seoul)	12:00–20:00 UTC
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

Code région	Nom de la région	Fenêtre de maintenance régionale
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
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 US East (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 du créneau de maintenance de votre cluster

Le créneau de maintenance doit intervenir au moment où l'utilisation est la plus faible et peut donc nécessiter d'être modifié de temps en temps. Vous pouvez modifier votre cluster en spécifiant une plage de temps de 24 heures au cours de laquelle toutes les opérations de maintenance demandées doivent avoir lieu. Toute modification de cluster en suspens ou différé demandée doit avoir lieu au cours de cette période.

Note

Si vous souhaitez appliquer les modifications de type de nœud et/ou les mises à niveau du moteur immédiatement à l'aide de la case AWS Management Console Sélectionnez Appliquer maintenant. Sinon, ces modifications seront appliquées au cours de votre prochaine fenêtre de maintenance planifiée. Pour utiliser l'API, consultez [modify-replication-group](#) ou [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](#) — Gestion du remplacement des nœuds
- [Modifier un ElastiCache cluster](#) — Modification du créneau de maintenance d'un cluster

Configuration des paramètres de moteur à l'aide de groupes de paramètres

Amazon ElastiCache utilise des paramètres pour contrôler les propriétés d'exécution de vos nœuds et de vos clusters. Habituellement, les dernières versions de moteurs comprennent des paramètres supplémentaires pour prendre en charge une fonctionnalité plus récente. Pour plus de détails sur les tableaux de paramètres, consultez [Paramètres spécifiques Memcached](#).

Bien entendu, certaines valeurs de paramètres telles que `maxmemory` sont déterminées par le type de nœud et de moteur. Pour un tableau des valeurs de ces paramètres de type de nœud, consultez [Paramètres propres au type de nœud Memcached](#).

 Note

Pour obtenir la liste des paramètres spécifiques à Memcached, consultez [Paramètres spécifiques Memcached](#).

Rubriques

- [Gestion des paramètres](#)
- [Niveaux de groupe de paramètres de cache](#)
- [Création d'un groupe de paramètres](#)
- [Liste des groupes de paramètres par nom](#)
- [Affichage des valeurs d'un groupe de paramètres](#)
- [Modification d'un groupe de paramètres](#)
- [Suppression d'un groupe de paramètres](#)
- [Paramètres spécifiques Memcached](#)

Gestion des paramètres

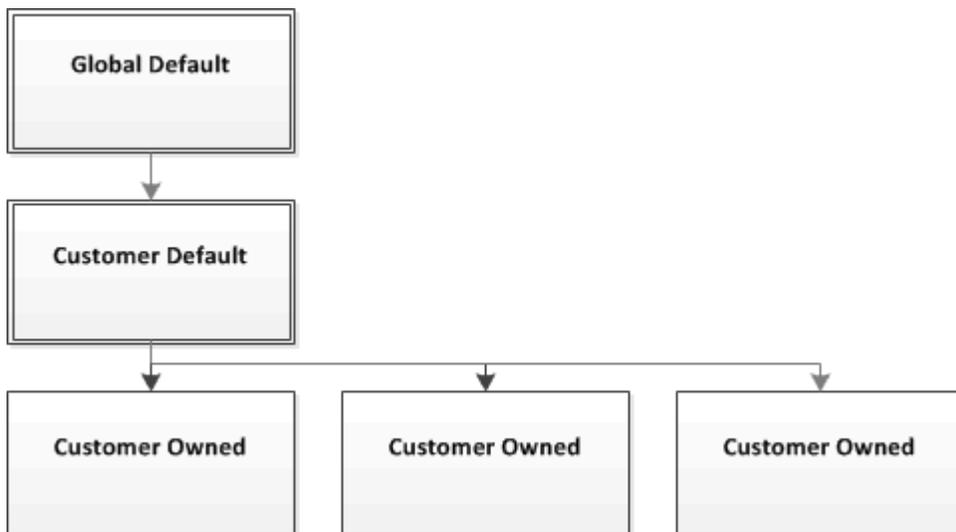
Les paramètres sont regroupés dans les groupes de paramètre nommés pour faciliter la gestion paramètre. Un groupe de paramètres représente une combinaison de valeurs spécifiques pour les paramètres qui sont transmis au logiciel de moteur de au moment du démarrage. Ces valeurs déterminent le comportement des processus du moteur sur chaque nœud au moment de l'exécution. Les valeurs des paramètres sur un groupe de paramètres spécifiques s'appliquent à tous les nœuds associés au groupe, indépendamment du cluster auquel ils appartiennent.

Pour affiner les performances de votre cluster, vous pouvez modifier certaines valeurs des paramètres ou modifier le groupe de paramètres du cluster.

- Vous ne pouvez pas modifier, ni supprimer les groupes de paramètres par défaut. Si vous avez besoin de valeurs des paramètres personnalisés, vous devez créer un groupe de paramètres personnalisés.
- La famille de groupe de paramètres et le cluster que vous lui associez doivent être compatibles. Par exemple, si votre cluster utilise Memcached version 1.4.8, vous pouvez utiliser uniquement des groupes de paramètres par défaut ou personnalisés de la famille Memcached 1.4.
- 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.
- Lorsque vous modifiez les paramètres d'un cluster, la modification est immédiatement appliquée au cluster. C'est vrai si vous changez le groupe de paramètres même du cluster ou une valeur de paramètre dans le groupe de paramètres du cluster. 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](#).

Niveaux de groupe de paramètres de cache

Amazon ElastiCache comporte trois niveaux de groupe de paramètres de cache, comme illustré ci-dessous.



Niveaux de groupe de paramètres Amazon ElastiCache

Par défaut global

Le groupe de paramètres racine de niveau supérieur pour tous les clients Amazon ElastiCache dans la région.

Le groupe de paramètres de cache par défaut global :

- Est réservé à ElastiCache et n'est pas disponible pour le client.

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 de plus amples informations, veuillez consulter [Paramètres spécifiques Memcached](#).

Création d'un groupe de paramètres

Vous devez créer un groupe de paramètres s'il existe une ou plusieurs valeurs de paramètre que vous voulez changer par rapport aux valeurs par défaut. Vous pouvez créer un groupe de paramètres à l'aide de la console ElastiCache, de la AWS CLI ou l'API ElastiCache.

Création d'un groupe de 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 console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 groupe de paramètres, telle que `memcached1.4`, définit les paramètres réels dans votre groupe de paramètres et 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 de paramètres](#).

Création d'un groupe de paramètres (AWS CLI)

Pour créer un groupe de paramètres à l'aide de l'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.

Exemple

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"  
  }  
}
```

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 de paramètres](#).

Pour de plus amples informations, veuillez consulter [create-cache-parameter-group](#).

Création d'un groupe de paramètres (API ElastiCache)

Pour créer un groupe de paramètres à l'aide de l'API ElastiCache utilisez l'action `CreateCacheParameterGroup` 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`.
- `Description` – Une description fourni par l'utilisateur pour le groupe de paramètres.

Exemple

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>
```

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 de paramètres](#).

Pour de plus amples informations, veuillez consulter [CreateCacheParameterGroup](#).

Liste des groupes de paramètres par nom

Vous pouvez dresser la liste des groupes de paramètres à l'aide de la console ElastiCache, de la AWS CLI ou l'API ElastiCache.

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 lister les groupes de paramètres à l'aide de la console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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.

Liste des groupes de paramètres par nom (AWS CLI)

Pour générer une liste des groupes de paramètres à l'aide de l'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.

Exemple

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"
    }
  ]
}
```

Exemple

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](#).

Liste des groupes de paramètres par nom (API ElastiCache)

Pour générer une liste des groupes de paramètres à l'aide de l'API ElastiCache API, utilisez l'action `DescribeCacheParameterGroups`. 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.

Exemple

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>
```

Exemple

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>
```

Pour de plus amples informations, veuillez consulter [DescribeCacheParameterGroups](#).

Affichage des valeurs d'un groupe de paramètres

Vous pouvez afficher les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la console ElastiCache, de la AWS CLI ou de l'API ElastiCache.

Affichage des valeurs d'un groupe de paramètres (console)

La procédure suivante montre comment afficher les paramètres et leurs valeurs pour un groupe de paramètres à l'aide de la console ElastiCache.

Pour afficher les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de la console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 afficher les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de l'AWS CLI, utilisez la commande `describe-cache-parameters`.

Exemple

L'exemple de code suivant affiche tous les paramètres 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
```

Pour de plus amples informations, veuillez consulter [describe-cache-parameters](#).

Liste des valeurs d'un groupe de paramètres (API ElastiCache)

Pour afficher les paramètres d'un groupe de paramètres et leurs valeurs à l'aide de l'API ElastiCache, utilisez l'action DescribeCacheParameters.

Exemple

L'exemple de code suivant affiche tous les paramètres du 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>
```

Pour de plus amples informations, veuillez consulter [DescribeCacheParameters](#).

Modification d'un groupe de paramètres

Important

Vous ne pouvez pas modifier un groupe de paramètres par défaut.

Vous pouvez modifier certaines valeurs des paramètres dans un groupe de paramètres. Ces valeurs de ces paramètres sont appliquées aux clusters associés au groupe de paramètres. Pour savoir quand une modification de valeur de paramètre est appliquée à un groupe de paramètres, consultez [Paramètres spécifiques Memcached](#).

Modification d'un groupe de paramètres

La procédure suivante montre comment modifier une valeur de paramètre `binding_protocol` à l'aide de la console ElastiCache. Vous devez utiliser la même procédure pour modifier la valeur de tout paramètre.

Pour modifier une valeur de paramètre à l'aide de la console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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. Sur l'écran Edit Parameter Group: faites défiler, à l'aide des flèches gauche et droite, pour rechercher le paramètre `cluster-enabled`, puis tapez `yes` dans la colonne Value.
7. Choisissez Enregistrer les modifications.
8. Pour rechercher 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](#).

Modification d'un groupe de paramètres (AWS CLI)

Pour modifier la valeur d'un paramètre à l'aide de l'AWS CLI, utilisez la commande `modify-cache-parameter-group`.

Exemple

Pour rechercher le nom du paramètre que vous avez modifié, ainsi que les valeurs autorisées, consultez [Paramètres spécifiques Memcached](#)

L'exemple de code suivant définit 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"  
}
```

Pour de plus amples informations, veuillez consulter [modify-cache-parameter-group](#).

Modification d'un groupe de paramètres (API ElastiCache)

Pour modifier les valeurs de paramètres d'un groupe de paramètres à l'aide de l'API ElastiCache, utilisez l'action `ModifyCacheParameterGroup`.

Exemple

Pour rechercher le nom du paramètre que vous avez modifié, ainsi que les valeurs autorisées, consultez [Paramètres spécifiques Memcached](#)

L'exemple de code suivant définit 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>
```

Pour de plus amples informations, veuillez consulter [ModifyCacheParameterGroup](#).

Suppression d'un groupe de paramètres

Vous pouvez supprimer un groupe de paramètres à l'aide de la console ElastiCache, de la AWS CLI ou de l'API ElastiCache.

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 console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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. Choisissez 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 l'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`.

Exemple

Pour Linux, macOS ou Unix :

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myMem14
```

Pour Windows :

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myMem14
```

Pour de plus amples informations, veuillez consulter [delete-cache-parameter-group](#).

Suppression d'un groupe de paramètres (API ElastiCache)

Pour supprimer un groupe de paramètres à l'aide de l'API ElastiCache, utilisez l'action `DeleteCacheParameterGroup`. 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.

Exemple

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>
```

Pour de plus amples informations, veuillez consulter [DeleteCacheParameterGroup](#).

Paramètres spécifiques 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 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. Vous ne pourrez pas activer/désactiver `lru_crawler` au moment de l'exécution via les commandes. Activez/désactivez `lru_crawler` en modifiant 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	<p>Valeur par défaut : 1</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Valeurs autorisées : 0,1</p> <p>Les modifications prennent effet : au lancement</p>	<p>Un alias pour désactiver les commandes <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> . No modern définit également le paramètre <code>hash_algorithm</code> sur <code>jenkins</code> et autorise l'inlining d'ASCII VALUE. Applicable à memcached 1.5 et versions ultérieures. Pour revenir à la version moderne, vous devez désactiver ce paramètre et le relancer, ce qui activera automatiquement <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , et <code>maxconns_fast</code> .</p> <div data-bbox="1008 1171 1511 1885" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>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 automatiquement récupérée par les nouveaux utilisateurs d'ElastiCache pour chaque région après le 20 août 2021. Les utilisateurs ElastiCache existants dans les régions avant</p></div>

Nom	Détails	Description
		le 20 août 2021 doivent modifier manuellement leurs groupes de paramètres personnalisés afin de récupérer cette nouvelle modification.
<code>inline_ascii_resp</code>	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 <code>get</code> et <code>fasten ASCII</code> .

Pour Memcached 1.5.10, les paramètres suivants sont supprimés.

Nom	Détails	Description
<code>expirezero_does_no_t_evict</code>	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	
<code>modern</code>	Valeur par défaut : 1 Type : valeur booléenne Modifiable : Oui (nécessite de relancer s'il est défini sur <code>no_modern</code>) Valeurs autorisées : 0,1 Les modifications prennent effet : au lancement	N'est plus pris en charge dans cette version. À partir de cette version, <code>no-modern</code> 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
<code>modern</code>	Par défaut : activé	Un alias pour plusieurs fonctions . Activer <code>modern</code> équivaut à

Nom	Détails	Description
	<p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>activer les commandes suivantes et à utiliser un algorithme de hachage murmur3 : <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> et <code>hash_algorithm=murmur3</code> .</p>
<code>watch</code>	<p>Par défaut : activé</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : imminent</p> <p>Des journaux sont supprimés si l'utilisateur atteint ses limites <code>watcher_logbuf_size</code> et <code>worker_logbuf_size</code> .</p>	<p>Extractions, expulsions ou mutations de journaux. Quand, par exemple, un utilisateur active <code>watch</code>, il peut voir des journaux lorsque <code>get</code>, <code>set</code>, <code>delete</code> ou <code>update</code> a lieu.</p>
<code>idle_timeout</code>	<p>Valeurs : 0, 1.</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>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.</p>

Nom	Détails	Description
<code>track_sizes</code>	<p>Par défaut : désactivé</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>Affiche les tailles consommées par chaque groupe de sections.</p> <p>Activer <code>track_sizes</code> vous permet d'exécuter <code>stats sizes</code> sans avoir à exécuter <code>stats sizes_enable</code> .</p>
<code>watcher_logbuf_size</code>	<p>Défaut : 256 (Ko)</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>La commande <code>watch</code> active la journalisation des flux pour Memcached. Cependant <code>watch</code> 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.</p>

Nom	Détails	Description
<code>worker_logbuf_size</code>	<p>Défaut : 64 (Ko)</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>La commande <code>watch</code> active la journalisation des flux pour Memcached. Cependant <code>watch</code> 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.</p>
<code>slab_chunk_max</code>	<p>Par défaut : 524288 (octets)</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>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 <code>slab_chunk_max</code> sont répartis sur plusieurs sections.</p>
<code>lru_crawler metadump [all 1 2 3]</code>	<p>Par défaut : désactivé</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : immédiatement</p>	<p>Si <code>lru_crawler</code> est activé, cette commande vide toutes les clés.</p> <p><code>all 1 2 3</code> - toutes les sections, ou spécifiez un numéro de section particulier</p>

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
<code>disable_flush_all</code>	<p>Valeurs : 0, 1.</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>Ajoutez un paramètre (-F) pour désactiver <code>flush_all</code>. Utile si vous ne souhaitez jamais effectuer de vidage complet de la mémoire sur les instances de production.</p> <p>Valeurs : 0, 1 (un utilisateur peut lancer un <code>flush_all</code> lorsque la valeur est 0).</p>
<code>hash_algorithm</code>	<p>Valeur par défaut : jenkins</p> <p>Type : chaîne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>L'algorithme de hachage à utiliser. Valeurs autorisées : <code>murmur3</code> et <code>jenkins</code>.</p>
<code>lru_crawler</code>	<p>Valeurs : 0, 1.</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	<p>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 actuellement de lancer une analyse à l'aide d'une commande manuelle.</p>

Nom	Détails	Description
	<p> Note</p> <p>Vous pouvez activer temporairement <code>lru_crawler</code> lors de l'exécution de la ligne de commande. Pour plus d'informations, consultez la colonne Description.</p>	<p>Pour activer temporairement, exécutez <code>lru_crawler enable</code> à la ligne de commande.</p> <p><code>lru_crawler 1,3,5</code> analyse les classes de sections 1, 3 et 5 à la recherche des éléments arrivés à expiration à ajouter à la freelist.</p> <p>Valeurs : 0,1</p> <p> Note</p> <p>Activer <code>lru_crawler</code> 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 de paramètres.</p>

Nom	Détails	Description
<code>lru_maintainer</code>	<p>Valeurs : 0, 1.</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>Un thread en arrière-plan qui mélange des éléments entre les LRU (éléments les moins utilisés) quand les capacités sont accessibles. Valeurs : 0, 1.</p>
<code>expirezero_does_not_evict</code>	<p>Valeurs : 0, 1.</p> <p>Type : valeur booléenne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : au lancement</p>	<p>Si utilisé avec le paramètre <code>lru_maintainer</code>, les éléments ayant une date d'expiration de 0 ne peuvent pas être supprimés.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Warning</p> <p>Cela évite d'utiliser la mémoire disponible pour les autres éléments pouvant être supprimés.</p> </div> <p>Peut être paramétré pour ne pas tenir compte de <code>lru_maintainer</code>.</p>

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
<code>config_max</code>	Le nombre maximum d'entrées de configuration ElastiCache.
<code>config_size_max</code>	La taille maximale des entrées configuration, en octets.
<code>hashpower_init</code>	La taille initiale de la table de hachage d'ElastiCache, exprimée en puissance de deux. La valeur par défaut est 16 (2^{16}) ou 65536 clés.
<code>maxconns_fast</code>	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
	<p>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 un message d'erreur au client et ferme immédiatement la connexion.</p>
slab_automove	<p>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 défini sur 2, ElastiCache déplace énergiquement les sections dès qu'une expulsion se produit. (Ce mode n'est pas recommandé sauf pour effectuer des tests).</p>

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
<code>binding_protocol</code>	<p>Valeur par défaut : auto</p> <p>Type : chaîne</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	<p>Le protocole de liaison.</p> <p>Les valeurs autorisées sont : <code>ascii</code> et <code>auto</code>.</p> <p>Pour une assistance sur la modification de la valeur de <code>binding_protocol</code> , consultez Modification d'un groupe de paramètres.</p>
<code>cas_disabled</code>	<p>Par défaut : 0 (faux)</p> <p>Type : booléen</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	<p>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.</p>
<code>chunk_size</code>	<p>Par défaut: 48</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	<p>Le montant minimum, en octets, de l'espace à allouer à la clé, à la valeur et aux indicateurs du plus petit élément.</p>
<code>chunk_size_growth_factor</code>	<p>Par défaut: 1.25</p> <p>Type : décimal</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	<p>Le facteur de croissance qui contrôle la taille de chaque segment Memcached successif ; chaque segment sera <code>chunk_size_growth_factor</code> fois plus important que le segment précédent.</p>

Nom	Détails	Description
<code>error_on_memory_exhausted</code>	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.
<code>large_memory_pages</code>	Par défaut : 0 (faux) Type : booléen Modifiable : non	Si 1 (vrai), ElastiCache essaie d'utiliser les pages de mémoire volumineuses.
<code>lock_down_paged_memory</code>	Par défaut : 0 (faux) Type : booléen Modifiable : non	Si 1 (vrai), ElastiCache verrouillera toute la mémoire paginée.
<code>max_item_size</code>	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.
<code>max_simultaneous_connections</code>	Par défaut: 65000 Type : entier Modifiable : non	Le nombre maximum de connexions simultanées.

Nom	Détails	Description
<code>maximize_core_file_limit</code>	<p>Par défaut : 0 (faux)</p> <p>Type : booléen</p> <p>Adaptabilité:</p> <p>Les modifications prennent effet : après le redémarrage</p>	Si 1 (vrai), ElastiCache optimisera la limite du fichier de base.
<code>memcached_connections_overhead</code>	<p>Par défaut : 100</p> <p>Type : entier</p> <p>Modifiable : oui</p> <p>Les modifications prennent effet : après le redémarrage</p>	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 .
<code>requests_per_event</code>	<p>Valeur par défaut : 20</p> <p>Type : entier</p> <p>Modifiable : non</p>	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, ElastiCache définit la valeur pour `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	<code>max_cache_memory</code> (en méga-octets)	<code>num_threads</code>
<code>cache.t1.micro</code>	213	1
<code>cache.t2.micro</code>	555	1
<code>cache.t2.small</code>	1588	1
<code>cache.t2.medium</code>	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).

Mise à l'échelle ElastiCache pour Memcached

Mise à l'échelle ElastiCache pour Memcached

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 du stockage des données du cache et de l'ECPU/seconde 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 une limite maximale de mise à l'échelle, votre application risque de voir les performances du cache diminuer lorsque celui-ci atteint la limite maximale. 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 commencent à être expulsées en utilisant la logique LRU. Lorsque vous définissez un maximum d'ECPU/seconde et que l'utilisation du calcul de votre charge de travail dépasse cette valeur, ElastiCache les requêtes Memcached 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 (eCPU) par seconde ou pour le stockage de 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 ce pic d'utilisation significatif.

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/seconde disponibles sur 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 d'ECPUS/seconde ou de stockage de 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 30 000 ECPUs/seconde sur un cache vide, et jusqu'à 90 000 ECPUs/s lors de l'utilisation de la fonction Read from Replica. ElastiCache peut doubler l'eCPU/seconde toutes les 10 à 12 minutes. Cette vitesse de mise à l'échelle est suffisante pour la plupart des charges de travail. Si vous prévoyez qu'un événement de dimensionnement à venir pourrait dépasser ce taux, nous vous recommandons de définir le nombre minimal d'eCPUs/seconde par rapport au pic d'eCPUs/seconde que vous attendez 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 le nouveau nombre minimum d'eCPU 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 ECPUs/seconde, vous serez facturé au moins 1,224\$ par heure (en utilisant les prix des

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 ECPU/seconde par slot (90 000 ECPU/seconde 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 Redis et le trafic entre les clés soient aussi uniformes que possible.

Définition des limites de dimensionnement 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/](https://console.aws.amazon.com/elasticache/).
2. Dans le volet de navigation, choisissez le moteur s'exécutant sur le cache que vous voulez 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-cache` API.

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 ElastiCache pour les clusters conçus par Memcached

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.

Ce qui suit vous permet de trouver la rubrique appropriée pour les actions de mise à l'échelle que vous souhaitez exécuter.

Mise à l'échelle des clusters Memcached

Action	Rubrique
Augmentation d'échelle	Ajout de nœuds à un cluster

Action	Rubrique
Diminution d'échelle	Suppression de nœuds d'un cluster
Modification des types de nœuds	Mise à l'échelle verticale de Memcached

Les clusters Memcached sont composés de 1 à 60 nœuds. La mise à l'échelle d'un cluster de Memcached est aussi simple que d'ajouter ou de supprimer des nœuds à partir du cluster.

Si vous avez besoin de plus de 60 nœuds dans un cluster Memcached, ou de plus de 300 nœuds au total dans une AWS région, remplissez le formulaire de demande d'augmentation de ElastiCache limite à l'[adresse https://aws.amazon.com/contact-us/elasticache-node-limit-request/](https://aws.amazon.com/contact-us/elasticache-node-limit-request/).

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, dans la mesure où le moteur Memcached ne stocke pas les données de façon durable, si vous passez à un type de nœud différent, votre nouveau cluster démarre à vide, sauf si votre application le renseigne.

Rubriques

- [Mise à l'échelle horizontale de Memcached](#)
- [Mise à l'échelle verticale de Memcached](#)

Mise à l'échelle horizontale de Memcached

Le moteur Memcached prend en charge le partitionnement de vos données sur plusieurs nœuds. En conséquence, il est facile de dimensionner horizontalement les clusters de Memcached. Un cluster Memcached peut comporter de 1 à 60 nœuds. Pour dimensionner horizontalement votre cluster de Memcached, ils suffit simplement d'ajouter ou de supprimer des nœuds.

Si vous avez besoin de plus de 60 nœuds dans un cluster Memcached, ou de plus de 300 nœuds au total dans une AWS région, remplissez le formulaire de demande d'augmentation de ElastiCache limite à l'[adresse https://aws.amazon.com/contact-us/elasticache-node-limit-request/](https://aws.amazon.com/contact-us/elasticache-node-limit-request/).

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 client ElastiCache pour un équilibrage de charge efficace](#).

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 dans votre cluster](#). 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.

Mise à l'échelle verticale de Memcached

Lorsque vous augmentez ou diminuez votre cluster Memcached, vous devez créer un nouveau cluster. Les clusters de Memcached commencent toujours vides, sauf si votre application les remplit.

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 plus d'informations, consultez [Sélection de la taille du nœud de cache](#).

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 [Recherche des points de terminaison d'un cluster \(console\)](#).
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 plus d'informations, consultez [Création de clusters avec l'interface de ligne de commande](#).
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 de AWS CLI](#).

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 plus d'informations, consultez [Création d'un cluster \(ElastiCache API\)](#).
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 des points de terminaison \(API ElastiCache\)](#).

3. Supprimez l'ancien cluster de cache. Pour de plus amples informations, veuillez consulter [Utilisation de l'API ElastiCache](#).

Étiquetage de vos ressources ElastiCache

Pour vous aider à gérer vos ressources ElastiCache, vous pouvez attribuer vos propres métadonnées à chaque ressource sous la forme de balises. Les balises vous permettent de classer vos ressources AWS 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. 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 balise est une étiquette que vous affectez à une ressource AWS. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Les balises vous permettent de classer vos ressources AWS de différentes manières, par exemple, par objectif ou par propriétaire. Par exemple, vous pouvez définir un ensemble de balises pour les clusters ElastiCache de votre compte, ce qui vous permet 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 pour ElastiCache et sont interprétées strictement comme une chaîne de caractères. De plus, les étiquettes ne sont pas automatiquement affectées à vos ressources. Vous pouvez modifier les clés et valeurs de balise, et vous pouvez retirer des balises d'une ressource à tout moment. Vous pouvez définir la valeur d'une balise à `null`. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur. Si vous supprimez une ressource, ses balises sont également supprimées. 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 gérer les balises à l'aide de la AWS Management Console, de la AWS CLI et de l'API ElastiCache.

Si vous utilisez IAM, vous pouvez contrôler quels utilisateurs de votre compte AWS sont autorisés à créer, modifier ou supprimer des étiquettes. Pour de plus amples informations, veuillez consulter [Autorisations de niveau ressource](#).

Ressources que vous pouvez étiqueter

Vous pouvez étiqueter à la plupart des ressources ElastiCache qui existent déjà dans votre compte. Le tableau ci-dessous répertorie les ressources qui prennent en charge le balisage. Si vous utilisez l'outil AWS Management Console, vous pouvez appliquer des balises aux ressources à l'aide de l'outil [Tag Editor \(Éditeur de balises\)](#). 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 des ressources en fonction de la balise Name (Nom), mais celle-ci n'a pas de signification sémantique pour le service ElastiCache.

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 scripts de balisage personnalisés après la création de ressources.

Si vous utilisez l'API Amazon ElastiCache, la CLI AWS ou un kit SDK AWS, vous pouvez appliquer les balises aux nouvelles ressources sur l'action d'API ElastiCache correspondante à l'aide du paramètre Tags. Il s'agit des options suivantes :

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `PurchaseReservedCacheNodesOffering`

Le tableau suivant décrit les ressources ElastiCache qui peuvent être étiquetées, et les ressources qui peuvent être étiquetées lors de la création à l'aide de l'API ElastiCache, de la CLI AWS ou d'un kit SDK AWS.

Prise en charge de l'étiquetage pour les ressources ElastiCache

Prend en charge les balises	Prend en charge le balisage au moment de la création
Oui	Oui

Vous pouvez appliquer des autorisations au niveau des ressources basées sur des balises dans vos politiques IAM aux actions d'API ElastiCache qui prennent en charge l'étiquetage à la création, afin de mettre en œuvre un contrôle détaillé des utilisateurs et des groupes qui peuvent étiqueter des ressources à leur 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](#).

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'ElastiCache autorise tous les caractères 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 de balise sont sensibles à la casse.
- Le préfixe `aws :` est réservé à l'utilisation d'AWS. 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 labelisés avec une clé de balise appelée `DeleteMe`, vous devez utiliser l'action `DeleteSnapshot` avec les identificateurs de ressource des instantanés, tels que `snap-1234567890abcdef0`.

Pour plus d'informations sur les ressources ElastiCache que vous pouvez étiqueter, veuillez consulter [Ressources que vous pouvez étiqueter](#).

Exemple : étiquetage de vos ressources

- Création d'un cache sans serveur à l'aide de balises

```
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-id <resource-id> \  
  --tags <tags>
```

```
--resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \
--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 memcached \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

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`.

```
{
  "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`.

```
{
```

```

"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`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:AddTagsToResource",
      "Resource": [
        "arn:aws:elasticache:*:*:*:*"
      ],
      "Condition": {
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [
            "Service",
            "Project"
          ]
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

4. Refus de l'action `CreateCacheCluster` si l'identification de la demande `Project` est manquante ou n'est pas égale à `Dev`, `QA` ou `Prod`.

```

{
  "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 dans Amazon ElastiCache, vous pouvez suivre les coûts en regroupant les dépenses sur vos factures par des valeurs de balise de ressource.

Une balise ElastiCache de répartition des coûts est une paire clé-valeur que vous définissez et associez avec une ressource ElastiCache. 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.

Vous pouvez utiliser des balises répartition des coûts pour organiser votre facture AWS afin de refléter votre propre structure de coût. Pour ce faire, inscrivez-vous pour obtenir votre facture de compte AWS avec les valeurs de clé de balise 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 le total de vos coûts ElastiCache réparti par région. Pour de plus amples informations, veuillez consulter [Utilisation des balises d'allocation des coûts](#) dans le Guide de l'utilisateur AWS Billing.

Vous pouvez ajouter des balises de répartition des coûts ElastiCache à des clusters Memcached. Lorsque vous ajoutez, affichez, modifiez, copiez ou supprimez une balise, l'opération est appliquée uniquement au cluster spécifié.

Caractéristiques des balises de répartition des coûts ElastiCache

- Des balises de répartition des coûts sont appliquées à des ressources ElastiCache qui sont spécifiées en tant qu'ARN dans les opérations de la CLI et de l'API. Le type de ressource sera un « cluster ».

Exemple d'ARN : `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Memcached : Les balises sont appliquées uniquement aux clusters.

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 ressource ElastiCache peut avoir 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 les balises sur tout type de ressource ElastiCache.

Gestion des balises de répartition des coûts l'aide de la AWS CLI

Vous pouvez utiliser l'AWS CLI pour ajouter, modifier ou supprimer des balises de répartition des coûts.

Les balises de répartition des coûts sont appliquées à ElastiCache pour des 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`

Exemple d'ARN : `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Rubriques

- [Répertoirer les balises à l'aide de la AWS CLI](#)
- [Ajout de balises à l'aide de la AWS CLI](#)
- [Modification des balises à l'aide de la AWS CLI](#)
- [Suppression de balises à l'aide de la AWS CLI](#)

Répertoirer les balises à l'aide de la AWS CLI

Vous pouvez utiliser la AWS CLI pour lister les balises sur une ressource ElastiCache existante en utilisant l'opération [list-tags-for-resource](#).

Le code suivant utilise l'AWS CLI pour afficher la liste des balises sur le 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 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"  
    }  
  ]  
}
```

En l'absence de balises sur la ressource, le résultat sera une TagList vide.

```
{  
  "TagList": []  
}
```

Pour plus d'informations, veuillez consulter la AWS CLI pour ElastiCache [list-tags-for-resource](#).

Ajout de balises à l'aide de la AWS CLI

Vous pouvez utiliser la AWS CLI pour ajouter des balises à une ressource ElastiCache existante en utilisant l'opération de la CLI [add-tags-to-resource](#). 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'AWS CLI pour ajouter les clés Service et Region avec les valeurs elasticache et us-west-2 respectivement au cluster my-cluster dans la région us-west-2.

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
```

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, veuillez consulter la AWS CLI pour ElastiCache [add-tags-to-resource](#).

Vous pouvez également utiliser l'AWS CLI pour ajouter des balises à un cluster lorsque vous créez un nouveau cluster à l'aide de l'opération [create-cache-cluster](#). Vous ne pouvez pas ajouter des balises lors de la création d'un cluster à l'aide de la console de gestion ElastiCache. 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 de la AWS CLI

Vous pouvez utiliser la AWS CLI pour modifier les balises sur un cluster ElastiCache for Memcached.

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 de la AWS CLI

Vous pouvez utiliser la AWS CLI pour supprimer des balises d'un cluster ElastiCache for Memcached existant en utilisant l'opération [remove-tags-from-resource](#).

Le code suivant utilise l'AWS CLI pour supprimer les balises avec les clés Service et Region du le cluster `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
```

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, veuillez consulter la AWS CLI pour ElastiCache [remove-tags-from-resource](#).

Gestion des balises de répartition des coûts l'aide de l'API ElastiCache

Vous pouvez utiliser l'API ElastiCache pour ajouter, modifier ou supprimer des balises de répartition des coûts.

Les balises de répartition des coûts sont appliquées à ElastiCache pour des 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 balises à l'aide de l'API ElastiCache](#)
- [Ajout de balises en utilisant l'API ElastiCache](#)
- [Modification des balises à l'aide de l'API ElastiCache](#)
- [Suppression de balises à l'aide de l'API ElastiCache](#)

Lister les balises à l'aide de l'API ElastiCache

Vous pouvez utiliser l'API ElastiCache pour lister les balises sur une ressource existante en utilisant l'opération [ListTagsForResource](#).

Le code suivant utilise l'API ElastiCache pour afficher la liste des balises sur la ressource `my-cluster` dans la région `usa-ouest-2` (`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>
```

Ajout de balises en utilisant l'API ElastiCache

Vous pouvez utiliser l'API ElastiCache pour ajouter des balises à un cluster ElastiCache existant en utilisant l'opération [AddTagsToResource](#). Si la clé de balise n'existe pas sur la ressource, la clé et la

valeur sont ajoutées à la ressource. Si la clé existe déjà sur la ressource, la valeur associée à cette clé est mise à jour en la nouvelle valeur.

Le code suivant utilise l'API ElastiCache pour ajouter les clés `Service` et `Region` avec les valeurs de `elasticache` et `us-west-2` respectivement à la ressource `my-cluster` dans la région `us-ouest-2` (`us-west-2`).

```
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>
```

Pour plus d'informations, voir [AddTagsToResource](#) dans la référence d'API Amazon ElastiCache.

Modification des balises à l'aide de l'API ElastiCache

Vous pouvez utiliser l'API ElastiCache pour modifier les balises sur un cluster ElastiCache.

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.

Suppression de balises à l'aide de l'API ElastiCache

Vous pouvez utiliser l'API ElastiCache pour supprimer des balises d'un cluster ElastiCache for Memcached existant en utilisant l'opération [RemoveTagsFromResource](#).

Le code suivant utilise l'API ElastiCache pour supprimer les balises avec les clés Service et Region du cluster my-cluster dans la région usa-ouest-2 (us-west-2).

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&TagKeys.member.1=Service  
&TagKeys.member.2=Region  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Utilisation du cadre Amazon ElastiCache Well-Architected

Cette section décrit le cadre Amazon ElastiCache Well-Architected, un ensemble de principes de conception et de conseils pour concevoir des charges de travail ElastiCache bien architecturées.

- Le cadre ElastiCache est un ajout à [AWS Well-Architected Framework](#).
- Chaque pilier comporte une série de questions destinées à lancer la discussion autour d'une architecture ElastiCache.
 - 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 ressources AWS 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 une valeur ajoutée 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.

Rubriques

- [Cadre Amazon ElastiCache Well-Architected - Pilier Excellence opérationnelle](#)
- [Cadre Amazon ElastiCache Well-Architected - Pilier Sécurité](#)
- [Cadre Amazon ElastiCache Well-Architected - Pilier Fiabilité](#)
- [Cadre Amazon ElastiCache Well-Architected - Pilier Efficacité des performances](#)
- [Cadre Amazon ElastiCache Well-Architected - Pilier Optimisation des coûts](#)

Cadre Amazon ElastiCache Well-Architected - Pilier Excellence opérationnelle

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

- [EO 1 : Comment comprenez-vous les alertes et les événements déclenchés par votre cluster ElastiCache et comment y répondez-vous ?](#)
- [EO 2 : Quand et comment mettez-vous à l'échelle vos clusters ElastiCache existants ?](#)
- [EO 3 : Comment gérez-vous les ressources de votre cluster ElastiCache et maintenez-vous votre cluster à jour ?](#)
- [EO 4 : Comment gérez-vous les connexions des clients à vos clusters ElastiCache ?](#)
- [EO 5 : Comment déployez-vous les composants ElastiCache pour une charge de travail ?](#)
- [EO 6 : Comment planifiez-vous et atténuez-vous les défaillances ?](#)
- [EO 7 : Comment résolvez-vous les problèmes liés aux événements du moteur Redis ?](#)

EO 1 : Comment comprenez-vous les alertes et les événements déclenchés par votre cluster ElastiCache et comment y répondez-vous ?

Introduction au niveau de la question : lorsque vous utilisez des clusters ElastiCache, vous pouvez éventuellement recevoir des notifications et des alertes lorsque des événements spécifiques se produisent. Par défaut, ElastiCache journalise les [événements](#) liés à vos ressources, tels qu'un basculement, le remplacement d'un nœud, une opération de mise à l'échelle, 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 par ElastiCache sur la console ElastiCache (après avoir sélectionné votre région) ou à l'aide de la commande [describe-events](#) de l'[interface de ligne de commande Amazon](#) (AWS CLI) et de l'[API ElastiCache](#). Configurez ElastiCache pour envoyer des notifications pour des événements de cluster importants à l'aide d'Amazon Simple Notification Service (Amazon SNS). L'utilisation d'Amazon SNS avec vos clusters vous permet de prendre des mesures par programmation en fonction des 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 (uniquement Redis)
- [Ressources] :
 - [Gestion des notifications ElastiCache Amazon SNS](#)
 - [Notifications d'événements Amazon SNS](#)
- [Meilleure pratique] 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 les métriques Amazon CloudWatch pour surveiller vos clusters.

[Ressources] : pour obtenir un cas d'utilisation de Lambda et SNS, consultez [Monitor Amazon ElastiCache for Redis \(cluster mode disabled\) read replica endpoints using AWS Lambda, Amazon Route 53, and Amazon SNS](#).

EO 2 : Quand et comment mettez-vous à l'échelle vos clusters ElastiCache existants ?

Introduction au niveau de la question : le dimensionnement correct de votre cluster ElastiCache est une question d'équilibre qui doit être évaluée chaque fois que des modifications sont apportées aux types de charges 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, ElastiCache peut effectuer deux types de mises à l'échelle. 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. Vous pouvez le configurer dans votre bibliothèque cliente Redis en vous connectant au point de terminaison du lecteur ElastiCache en mode cluster désactivé, ou en utilisant la commande Redis READONLY en mode cluster activé.

[Ressources] :

- [Recherche de points de terminaison de connexion](#)
- [Cluster Right-Sizing](#)
- [Redis READONLY Command](#)
- [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 cluster d'ElastiCache for Redis est désactivé, le nœud primaire et le nœud de réplicas peuvent être mis à l'échelle 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] :

- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Mise à l'échelle des clusters ElastiCache for Redis](#)
- [Mise à l'échelle des clusters ElastiCache for 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 les métriques CloudWatch pour effectuer des analyses sur des plages temporelles plus longues. Les enseignements tirés de l'observation des métriques CloudWatch sur de longues périodes devraient éclairer vos prévisions concernant l'utilisation des ressources du cluster. Les points de données et les métriques CloudWatch sont disponibles pendant 455 jours au maximum.

[Ressources] :

- [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)
- [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)
- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Meilleure pratique] Si vos ressources ElastiCache sont créées avec CloudFormation, il est recommandé d'effectuer des modifications à l'aide de modèles CloudFormation afin de préserver la cohérence opérationnelle et d'éviter les modifications de configuration non gérées et les écarts de pile.

[Ressources] :

- [ElastiCache resource type reference for CloudFormation](#)
- [Meilleure pratique] Automatisez vos opérations de mise à l'échelle à l'aide des données opérationnelles du cluster et définissez des seuils dans CloudWatch pour configurer des alarmes. Utilisez CloudWatch Events et Simple Notification Service (SNS) pour déclencher des fonctions Lambda et exécuter une API ElastiCache afin de mettre à l'échelle vos clusters automatiquement. 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'alarmes Amazon CloudWatch](#)
- [Qu'est-ce qu'Amazon CloudWatch Events ?](#)
- [Utilisation d'Amazon SNS avec un kit SDK AWS Lambda](#)
- [Référence d'API ElastiCache](#)

EO 3 : Comment gérez-vous les ressources de votre cluster ElastiCache et maintenez-vous votre cluster à jour ?

Introduction au niveau de la question : lorsque vous travaillez à grande échelle, il est essentiel que vous soyez en mesure de localiser et d'identifier toutes vos ressources ElastiCache. Lorsque vous déployez de nouvelles fonctionnalités d'application, vous devez créer une symétrie des versions du cluster dans tous vos types d'environnements ElastiCache : 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. En tirant parti des fonctionnalités natives d'ElastiCache, votre équipe d'ingénieurs peut se concentrer sur la réalisation des objectifs commerciaux et non sur la maintenance d'ElastiCache.

- [Meilleure pratique] Exécutez 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 fenêtre 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 de services dans ElastiCache for Redis](#)
- [Page d'aide sur la maintenance gérée et les mises à jour de service d'Amazon ElastiCache](#)
- [Meilleure pratique] Organisez vos ressources ElastiCache 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 resource type reference for CloudFormation](#)
- [Groupes de paramètres](#)

EO 4 : Comment gérez-vous les connexions des clients à vos clusters ElastiCache ?

Introduction au niveau de la question : lorsque vous travaillez à grande échelle, vous devez comprendre comment vos clients se connectent au cluster ElastiCache pour gérer les aspects opérationnels de votre application (comme 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 répliques pour exécuter les opérations de lecture. Toutefois, sachez 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 Redis. La commande WAIT peut être utilisée pour améliorer la sécurité des données réelles et forcer les répliques à accuser réception des écritures avant de répondre aux clients, moyennant un coût global en termes de performances. L'utilisation de nœuds de répliques pour les opérations de lecture peut être configurée dans votre bibliothèque cliente ElastiCache for Redis en utilisant le point de terminaison du lecteur ElastiCache en mode cluster désactivé. En mode cluster activé, utilisez la commande READONLY d'ElastiCache for Redis. Pour la plupart des bibliothèques clientes d'ElastiCache for Redis, la commande READONLY d'ElastiCache for Redis est implémentée par défaut ou via un paramètre de configuration.

[Ressources] :

- [Recherche de points de terminaison de connexion](#)
- [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 cliente ElastiCache for Redis (si elle est prise en charge), avec un Framework disponible pour votre environnement d'applications 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] : [Configure Amazon ElastiCache for Redis for higher availability](#)

- [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](#)

EO 5 : Comment déployez-vous les composants ElastiCache pour une charge de travail ?

Introduction au niveau de la question : les environnements ElastiCache peuvent être déployés manuellement via la console AWS ou par programmation via des API, l'interface de ligne de commande, 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 clusters ElastiCache peuvent être isolés par charge de travail ou combinés à des fins d'optimisation des coûts.

Avantage au niveau de la question : le choix du mécanisme de déploiement le plus approprié pour vos environnements ElastiCache 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 de disposer d'environnements ElastiCache dédiés par charge de travail, de combiner plusieurs charges de travail dans des clusters uniques ou d'utiliser une combinaison des deux mécanismes. Comprendre les compromis peut aider à trouver un équilibre entre excellence opérationnelle et optimisation des coûts.

- [Obligatoire] Découvrez les options de déploiement disponibles pour ElastiCache et automatisez ces procédures dans la mesure du possible. Les pistes d'automatisation possibles incluent CloudFormation, l'interface de ligne de commande/le kit SDK AWS et les API.

[Ressources] :

- [Amazon ElastiCache resource type reference](#)
- [elasticache](#)
- [Référence d'API Amazon ElastiCache](#)
- [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 ressources ElastiCache 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 de la question : l'excellence opérationnelle consiste à anticiper les défaillances en effectuant régulièrement des exercices « pré-mortem » afin d'identifier les sources potentielles de défaillance dans le but de les éliminer ou de les atténuer. ElastiCache propose une API Failover qui permet de simuler des événements de défaillance de nœuds 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 sur les comptes de développement/test. [TestFailover](#)

EO 7 : Comment résolvez-vous les problèmes liés aux événements du moteur Redis ?

Introduction au niveau de la question : l'excellence opérationnelle exige de pouvoir étudier à la fois les informations au niveau du service et au niveau du moteur afin d'analyser l'état et le statut de vos clusters. Amazon ElastiCache for Redis peut transmettre les journaux du moteur Redis à Amazon CloudWatch et à Amazon Kinesis Data Firehose.

Avantage au niveau de la question : l'activation des journaux du moteur Redis sur les clusters Amazon ElastiCache for Redis fournit des informations sur les événements qui ont un impact sur l'état et les performances des clusters. Les journaux du moteur Redis fournissent des données provenant directement du moteur Redis qui ne sont pas disponibles via le mécanisme d'événements ElastiCache. En observant attentivement les événements ElastiCache (voir EO 1 précédent) et les journaux du moteur Redis, il est possible de déterminer l'ordre des événements lors de la résolution des problèmes, à la fois du point de vue du service ElastiCache et du point de vue du moteur Redis.

- [Obligatoire] Assurez-vous que la fonctionnalité de journalisation du moteur Redis est activée. Elle est disponible à partir d'ElastiCache for Redis 6.2 et 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.

- Sélectionnez un journal cible approprié dans CloudWatch ou 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?](#)
- [Meilleure pratique] Si vous utilisez Amazon CloudWatch Logs, envisagez de tirer parti d'Amazon CloudWatch Logs Insights pour interroger le journal du moteur Redis afin d'obtenir des informations importantes.

Par exemple, créez une requête sur le groupe de journaux CloudWatch contenant les journaux du moteur Redis qui renverra les événements dont le LogLevel est « WARNING ». Par exemple :

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Ressources] : [Analyse des données de journaux avec CloudWatch Logs Insights](#)

Cadre Amazon ElastiCache Well-Architected - Pilier Sécurité

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

- [PS 1 : Quelles mesures prenez-vous pour contrôler l'accès autorisé aux données ElastiCache ?](#)
- [PS 2 : Vos applications exigent-elles des autorisations supplémentaires pour accéder à ElastiCache en plus 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 ?](#)

- [PS 4 : Comment assurez-vous le chiffrement des données au repos avec ElastiCache ?](#)
- [PS 5 : Comment chiffrez-vous les données en transit avec 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 ?](#)

PS 1 : Quelles mesures prenez-vous pour contrôler l'accès autorisé aux données ElastiCache ?

Introduction au niveau de la question : tous les clusters ElastiCache 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 observé consiste à accéder à un cluster ElastiCache à partir d'une instance Amazon Elastic Compute Cloud dans le même VPC Amazon (Amazon Virtual Private Cloud). Avant de vous connecter à un cluster à partir d'une instance Amazon EC2, vous devez autoriser l'instance Amazon EC2 à accéder au cluster. Pour accéder à un cluster ElastiCache s'exécutant dans un VPC, il est nécessaire d'accorder l'entrée 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 en tant que pare-feu virtuel pour vos instances Amazon EC2 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 d'ElastiCache, lors du lancement d'un cluster, l'association d'un groupe de sécurité est requise. Cela garantit que les règles de trafic entrant et sortant sont en place pour tous les nœuds qui composent le cluster. En outre, ElastiCache 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 de trafic entrant n'est définie pour un groupe de sécurité et, par conséquent, aucun chemin d'entrée vers ElastiCache. Configurez une règle de trafic entrant sur le groupe de sécurité en spécifiant l'adresse/la plage IP source, le trafic de type TCP et le port de votre cluster ElastiCache (par exemple, le port par défaut est 6379 pour ElastiCache for Redis). 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 de trafic entrant, en autorisant par exemple uniquement l'accès entrant aux clients Redis s'exécutant sur des instances Amazon EC2 associées à un groupe de sécurité spécifique.

[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](#)
- [Obligatoire] Il est possible d'attribuer des politiques AWS Identity and Access Management aux fonctions AWS Lambda pour leur permettre d'accéder aux données ElastiCache. Pour activer cette fonctionnalité, créez un rôle d'exécution IAM avec l'autorisation `AWSLambdaVPCLambdaAccessExecutionRole`, puis attribuez le rôle à la fonction AWS Lambda.

[Ressources] : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un VPC Amazon : [Didacticiel : Configuration d'une fonction Lambda pour accéder à Amazon ElastiCache dans un VPC Amazon](#).

PS 2 : Vos applications exigent-elles des autorisations supplémentaires pour accéder à ElastiCache en plus des contrôles basés sur le réseau ?

Introduction au niveau de la question : dans les scénarios où il est nécessaire de restreindre ou de contrôler l'accès aux clusters ElastiCache for Redis au niveau d'un client individuel, il est recommandé de s'authentifier via la commande AUTH d'ElastiCache for Redis. Les jetons d'authentification ElastiCache for Redis, avec gestion facultative des utilisateurs et des groupes d'utilisateurs, permettent à ElastiCache for Redis d'exiger un mot de passe avant d'autoriser les clients à exécuter des commandes et des clés d'accès, améliorant ainsi la sécurité du plan de données.

Avantage au niveau de la question : pour garantir la sécurité de vos données, ElastiCache for Redis fournit des mécanismes de protection contre les accès non autorisés à vos données. Cela inclut l'application du contrôle d'accès basé sur les rôles (RBAC), ou du jeton AUTH (mot de passe) que les clients doivent utiliser pour se connecter à ElastiCache avant d'exécuter des commandes autorisées.

- [Meilleure pratique] Pour ElastiCache for Redis 6.x et ultérieures, définissez les contrôles d'authentification et d'autorisation en définissant des groupes d'utilisateurs, des utilisateurs et des 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 Redis qui prend en charge le protocole TLS pour pouvoir tirer parti du RBAC.

[Ressources] :

- [Application de RBAC à un groupe de réplication pour ElastiCache for Redis](#)
- [Définition des autorisations à l'aide d'une chaîne d'accès](#)
- [ACL](#)
- [ElastiCache pris en charge pour les versions Redis](#)
- [Meilleure pratique] Pour les versions d'ElastiCache for Redis antérieures à 6.x, outre la définition d'un jeton/mot de passe fort et le maintien d'une politique de mot de passe stricte pour ElastiCache for Redis AUTH, la meilleure pratique consiste à alterner 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 ElastiCache for Redis existant](#)

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 de la question : un certain nombre de commandes Redis 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 de la question : à partir d'ElastiCache for Redis 5.0.3 sur ElastiCache, vous pouvez 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 for Redis version 5.0.3 \(obsolète, utilisez la version 5.0.6\)](#)
- [Modifications des paramètres de Redis 5.0.3](#)
- [Redis security](#)

PS 4 : Comment assurez-vous le chiffrement des données au repos avec ElastiCache ?

Introduction au niveau de la question : ElastiCache for Redis est un stockage de données en mémoire, mais il est possible de chiffrer toutes les données susceptibles d'être conservées (dans 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 de la question : ElastiCache for Redis propose un chiffrement au repos facultatif pour renforcer la sécurité des données.

- [Obligatoire] Le chiffrement au repos ne peut être activé sur un cluster ElastiCache (groupe de réplication) que lorsqu'il est créé. Un cluster existant ne peut pas être modifié pour commencer à chiffrer les données au repos. Par défaut, ElastiCache fournit et gère les clés utilisées pour le chiffrement au repos.

[Ressources] :

- [Conditions du chiffrement au repos](#)
- [Activation du chiffrement au repos](#)
- [Meilleure pratique] Tirez parti des types d'instances Amazon EC2 qui chiffrent les données lorsqu'elles sont en mémoire (tels que 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 client (CMK). Grâce à l'intégration d'ElastiCache à AWS Key Management Service, vous pouvez créer, posséder et gérer les clés utilisées pour le chiffrement des données au repos pour votre cluster ElastiCache for Redis.

[Ressources] :

- [Utilisation des clés gérées par le client depuis AWS Key Management Service](#)
- [AWS Key Management Service](#)
- [AWS KMS concepts](#)

PS 5 : Comment chiffrez-vous les données en transit avec ElastiCache ?

Introduction au niveau de la question : il est souvent exigé d'éviter que les données ne soient compromises pendant leur transit. Cette exigence s'applique aux données au sein des composants d'un système distribué, ainsi qu'entre les clients d'application et les nœuds du cluster. ElastiCache for Redis répond à cette exigence en permettant de chiffrer les 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 au niveau de la question : le chiffrement en transit Amazon ElastiCache est une fonction facultative vous permettant d'améliorer la sécurité de vos données à leurs états les plus vulnérables, lorsqu'elles sont transférées d'un emplacement à un autre.

- [Obligatoire] Le chiffrement en transit ne peut être activé sur un cluster ElastiCache for Redis (groupe de réplication) que 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'activation du chiffrement en transit.

[Ressources] :

- [Présentation du chiffrement en transit](#)

PS 6 : Comment restreignez-vous l'accès aux ressources du plan de contrôle ?

Introduction au niveau de la question : les politiques IAM et l'ARN permettent des contrôles d'accès précis pour ElastiCache for Redis, ce qui permet de renforcer les contrôles pour gérer la création, la modification et la suppression des clusters ElastiCache for Redis.

Avantage au niveau de la question : la gestion des ressources Amazon ElastiCache, telles que les groupes de réplication, les nœuds, etc., peut être limitée aux comptes AWS disposant d'autorisations spécifiques basées sur des politiques IAM, ce qui améliore la sécurité et la fiabilité des ressources.

- [Obligatoire] Gérez l'accès aux ressources Amazon ElastiCache en attribuant des politiques AWS Identity and Access Management spécifiques aux utilisateurs AWS, ce qui permet de contrôler plus précisément quels comptes peuvent effectuer quelles actions sur les clusters.

[Ressources] :

- [Présentation de la gestion des autorisations d'accès à vos ressources ElastiCache](#)

- [Utilisation des 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 de la question : Lorsqu'il est déployé avec le RBAC activé, ElastiCache exporte les métriques CloudWatch 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, les ressources AWS relatives aux produits et services contribuent à sécuriser votre charge de travail globale en automatisant les déploiements et en journalisant toutes les actions et modifications pour un examen ou un audit ultérieur.

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 métriques CloudWatch publiées, relatives aux échecs d'authentification et d'autorisation RBAC.
 - AuthenticationFailures = Tentatives infructueuses d'authentification auprès de Redis
 - KeyAuthorizationFailures = Tentatives infructueuses d'accès à des clés sans autorisation
 - CommandAuthorizationFailures = Tentatives infructueuses d'exécution de commandes sans autorisation

[Ressources] :

- [Métriques pour Redis](#)
- [Meilleure pratique] Il est recommandé de configurer des alertes et des notifications sur ces métriques et d'y répondre si nécessaire.

[Ressources] :

- [Utilisation d'alarmes Amazon CloudWatch](#)
- [Meilleure pratique] Utilisez la commande Redis ACL LOG pour recueillir des détails supplémentaires.

[Ressources] :

- [ACL LOG](#)
- [Meilleure pratique] Familiarisez-vous avec les fonctionnalités AWS des produits et services en matière de surveillance, de journalisation et d'analyse des déploiements et des événements ElastiCache.

[Ressources] :

- [Journalisation des appels d'API Amazon ElastiCache avec AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-backup-check](#)
- [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)

Cadre Amazon ElastiCache Well-Architected - Pilier Fiabilité

Rubriques

- [PF 1 : Comment prenez-vous en charge les déploiements d'architecture haute disponibilité ?](#)
- [PF 2 : Comment atteignez-vous vos objectifs de point de reprise \(RPO\) avec ElastiCache ?](#)
- [PF 3 : Comment répondez-vous aux exigences de reprise après sinistre \(DR\) ?](#)
- [PF 4 : Comment planifiez-vous efficacement les basculements ?](#)
- [PF 5 : Vos composants ElastiCache sont-ils conçus pour être mis à l'échelle ?](#)

PF 1 : Comment prenez-vous en charge les déploiements d'architecture haute disponibilité ?

Introduction au niveau de la question : la compréhension de l'architecture haute disponibilité d'Amazon ElastiCache vous permettra de fonctionner de manière résiliente lors d'événements de disponibilité.

Avantage au niveau de la question : la configuration de vos clusters ElastiCache de manière à les rendre résilients aux pannes garantit une meilleure disponibilité de vos déploiements ElastiCache.

- [Obligatoire] Déterminez le niveau de fiabilité dont vous avez besoin pour votre cluster ElastiCache. 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 : Memcached ou ElastiCache for Redis

1. Memcached ne fournit aucun mécanisme de réplication et est principalement utilisé pour les charges de travail éphémères.
 2. ElastiCache for Redis propose les fonctionnalités haute disponibilité décrites ci-dessous.
- [Meilleure pratique] Pour les charges de travail nécessitant une haute disponibilité, utilisez ElastiCache for Redis en mode cluster avec au moins deux réplicas par partition, même pour les charges de travail exigeant 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 récupération plus rapide en cas de basculement, car le protocole Cluster de Redis exige que la majorité des nœuds primaires 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).

Amazon ElastiCache for Redis a optimisé les performances 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 dimensionnez correctement pour faire face aux pics de trafic anticipé : en cas de forte charge, le moteur ElastiCache for Redis peut ne plus répondre, ce qui affecte sa disponibilité. `BytesUsedForCache` et `DatabaseMemoryUsagePercentage` sont de bons indicateurs de votre utilisation de la mémoire, alors que `ReplicationLag` indique 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] :

- [Configure Amazon ElastiCache for Redis for higher availability](#)
- [Haute disponibilité avec les groupes de réplication](#)

PF 2 : Comment atteignez-vous vos objectifs de point de reprise (RPO) avec ElastiCache ?

Introduction au niveau de la question : comprenez à quoi correspond le RPO de la charge de travail pour prendre des décisions éclairées concernant les stratégies de sauvegarde et de récupération d'ElastiCache.

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 reprise (RPO) pour vos données ElastiCache. ElastiCache for Redis propose des fonctionnalités d'instantanés qui sont 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 déploiements ElastiCache.
 - Sachez que Memcached ne propose aucun processus de sauvegarde.
 - Passez en revue les fonctionnalités de sauvegarde et de restauration d'ElastiCache.
- [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.
- [Bonne pratique] Tirez parti de la fonctionnalité de sauvegarde planifiée d'ElastiCache pour sauvegarder régulièrement vos données pendant une fenêtre définie.
 - Testez régulièrement les restaurations à partir de vos sauvegardes.

- [Ressources] :
 - [Redis](#)
 - [Backup et restauration d'ElastiCache for Redis](#)
 - [Réalisation de sauvegardes manuelles](#)
 - [Planification des sauvegardes automatiques](#)
 - [Backup and Restore ElastiCache Redis Clusters](#)

PF 3 : Comment répondez-vous aux exigences de reprise après sinistre (DR) ?

Introduction au niveau de la question : la reprise après sinistre est un aspect important de toute planification de la charge de travail. ElastiCache for Redis propose plusieurs options pour implémenter la reprise après sinistre en fonction des exigences de résilience des charges de travail. Avec l'entrepôt de données global d'Amazon ElastiCache for Redis, vous pouvez écrire dans votre cluster ElastiCache for Redis dans une région et rendre les données disponibles en lecture à partir de deux autres clusters de réplicas entre régions, ce qui permet des lectures à faible latence et une reprise après sinistre entre 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 composants ElastiCache 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 sont 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. Veillez à effectuer le déploiement en ayant activé le mode Cluster dans les architectures multi-AZ, avec un minimum de 3 zones de disponibilité disponibles.
- 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 les régions AWS à l'aide d'entrepôts de données globaux](#)
 - [Restauration à partir d'une sauvegarde avec redimensionnement facultatif du cluster](#)
 - [Réduction des temps d'arrêt dans ElastiCache for Redis avec Multi-AZ](#)

PF 4 : Comment planifiez-vous efficacement les basculements ?

Introduction au niveau de la question : l'activation de Multi-AZ avec basculements automatiques est une meilleure pratique pour ElastiCache. Dans certains cas, ElastiCache for Redis remplace les nœuds primaires dans le cadre des opérations de maintenance. 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 des basculements dépend à la fois d'ElastiCache et de la configuration de votre bibliothèque cliente.

Avantage au niveau de la question : en suivant les meilleures pratiques relatives aux basculements d'ElastiCache conjointement avec votre bibliothèque cliente ElastiCache for Redis spécifique, vous pourrez minimiser les interruptions potentielles en cas 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 cliente ElastiCache for Redis, 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] :

- [Réduction des temps d'arrêt dans ElastiCache for Redis avec Multi-AZ](#)
- Passez en revue les meilleures pratiques de votre bibliothèque cliente ElastiCache for Redis.

- [Obligatoire] La réussite des basculements dépend de l'intégrité de l'environnement de réplication entre le nœud primaire et le nœud de réplica. Passez en revue et comprenez la nature asynchrone de la réplication Redis, ainsi que les métriques CloudWatch disponibles pour établir un rapport sur le décalage de réplication entre le nœud primaire et le nœud de réplica. Pour les cas d'utilisation nécessitant une meilleure sécurité des données, utilisez la commande Redis WAIT pour forcer les réplicas à accuser réception des écritures avant de répondre aux clients connectés.

[Ressources] :

- [Métriques pour Redis](#)
- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Meilleure pratique] Validez régulièrement la réactivité de votre application lors du basculement à l'aide de l'API Test Failover d'ElastiCache.

[Ressources] :

- [Testing Automatic Failover to a Read Replica on Amazon ElastiCache for Redis](#)
- [Test du basculement automatique](#)

PF 5 : Vos composants ElastiCache sont-ils conçus pour être mis à l'échelle ?

Introduction au niveau de la question : en comprenant les fonctionnalités de mise à l'échelle et les topologies de déploiement disponibles, vos composants ElastiCache peuvent s'adapter au fil du temps pour répondre à l'évolution des exigences en matière de charge de travail. ElastiCache propose 4 méthodes de mise à l'échelle : diminution/montée en puissance (horizontale) et augmentation/diminution de capacité (verticale).

Avantage au niveau de la question : le respect des meilleures pratiques pour les déploiements d'ElastiCache offre 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.
- [Meilleure pratique] Déployez vos composants ElastiCache avec le mode Cluster activé, en privilégiant un plus grand nombre de nœuds plus petits plutôt que des nœuds moins nombreux et plus volumineux. 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.
- [Bonne pratique] Si le mode cluster est désactivé, utilisez des réplicas en lecture pour augmenter la capacité globale en lecture. ElastiCache prend en charge jusqu'à 5 réplicas en lecture lorsque le mode cluster est désactivé, ainsi que la mise à l'échelle verticale.
- [Ressources] :
 - [Mise à l'échelle des clusters ElastiCache for Redis](#)
 - [Augmentation en ligne](#)
 - [Mise à l'échelle des clusters ElastiCache for Memcached](#)

Cadre Amazon ElastiCache Well-Architected - Pilier Efficacité des performances

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

- [EP 1 : Comment surveillez-vous les performances de votre cluster Amazon ElastiCache ?](#)
- [EP 2 : Comment répartissez-vous le travail entre vos nœuds de cluster ElastiCache ?](#)
- [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 ?](#)
- [EP 5 : Comment gérez-vous la suppression et/ou l'expulsion de clés ?](#)

- [EP 6 : Comment modélisez-vous et interagissez-vous avec les données dans ElastiCache ?](#)
- [EP 7 : Comment enregistrez-vous les commandes s'exécutant lentement dans votre cluster Amazon ElastiCache ?](#)
- [EP 8 : Comment l'autoscaling contribue-t-il à améliorer les performances du cluster ElastiCache ?](#)

EP 1 : Comment surveillez-vous les performances de votre cluster Amazon ElastiCache ?

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 métriques CloudWatch lors de l'exécution de ces tests afin de mieux comprendre les métriques disponibles et d'établir une référence en matière de performances.
- [Meilleure pratique] Pour les charges de travail ElastiCache for Redis, renommez les commandes coûteuses en ressources informatiques, comme KEYS par exemple, pour limiter la capacité des utilisateurs à exécuter des commandes de blocage sur les clusters de production.
 - Les charges de travail ElastiCache for Redis exécutant le moteur 6.x 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 console ou de l'interface de ligne de commande AWS et en associant les groupes d'utilisateurs à un cluster ElastiCache for Redis. Dans Redis 6, lorsque le RBAC est activé, « -@dangerous » peut être utilisé. Il interdit 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 paramètre `rename-commands` sur le groupe de paramètres du cluster Amazon ElastiCache for Redis.

- [Pratique encore meilleure] Analysez les requêtes lentes et recherchez des techniques d'optimisation.
 - Pour les charges de travail ElastiCache for Redis, apprenez-en plus sur vos requêtes en analysant le journal lent. Par exemple, vous pouvez utiliser la commande `redis-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 en utilisant des structures de données ElastiCache for Redis complexes. 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.
 - Pour les charges de travail ElastiCache for Redis, `redis-benchmark` fournit une interface simple permettant de 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 métriques CloudWatch](#)
 - [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)
 - [Utilisation d'alarmes Amazon CloudWatch](#)
 - [Paramètres spécifiques Redis](#)
 - [SLOWLOG](#)
 - [Redis benchmark](#)

EP 2 : Comment répartissez-vous le travail entre vos nœuds de cluster ElastiCache ?

Introduction au niveau de la question : la façon dont votre application se connecte aux nœuds Amazon ElastiCache peut avoir un impact sur les performances et la capacité de mise à l'échelle 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 terminaison ElastiCache approprié.
 - Amazon ElastiCache for Redis implémente différents points de terminaison en fonction du mode cluster utilisé. Lorsque le mode cluster est activé, ElastiCache fournit 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 lecture pour équilibrer les lectures entre les réplicas. 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 plusieurs nœuds, 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](#).
- [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 Redis multidimensionnelles telles que les listes, les flux, les ensembles, etc. Ces structures de données sont stockées dans des clés Redis 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. Pour les charges de travail ElastiCache for Redis, vous pouvez détecter les touches de raccourci en utilisant `redis-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 exige que le client écrive sur plusieurs nœuds primaires (le nœud Redis lui-même ne fournit pas cette fonctionnalité) et qu'il conserve la liste des noms de clés à lire, en plus du nom de clé d'origine.

- ElastiCache for Redis version 6 prend en charge la [mise en cache côté client](#) assistée par serveur. Cette fonctionnalité permet aux applications d'attendre les modifications apportées à une clé avant d'effectuer des appels réseau vers ElastiCache.
- [Ressources] :
 - [Configure Amazon ElastiCache for Redis for higher availability](#)
 - [Recherche de points de terminaison de connexion](#)
 - [Bonnes pratiques en matière d'équilibrage de charge](#)
 - [Client-side caching in Redis](#)

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 de la question : la mise en cache étant une charge de travail courante sur ElastiCache, 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] :

- [Obligatoire] Mesurez et collectez les performances du cache de votre application conjointement avec les valeurs de latence et d'utilisation de l'UC, afin de déterminer si vous devez ajuster votre durée de vie ou d'autres composants de l'application. ElastiCache fournit un ensemble de métriques CloudWatch pour les latences agrégées pour chaque structure de données. Ces métriques de latence sont calculées à l'aide de la statistique `commandstats` de la commande `INFO`

d'ElastiCache for Redis et n'incluent pas le temps réseau ni le temps d'E/S. Il s'agit uniquement du temps consommé par ElastiCache for Redis pour traiter les opérations.

[Ressources] :

- [Monitoring best practices with Amazon ElastiCache for Redis using 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](#)
- [Caching Best Practices](#)
- [Performance at Scale with Amazon ElastiCache Whitepaper](#)

EP 4 : Comment votre charge de travail optimise-t-elle l'utilisation des ressources de mise en réseau et des connexions ?

Introduction au niveau de la question : ElastiCache for Redis et Memcached sont pris en charge par de nombreux clients d'application 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 de manière proactive les connexions à votre cluster ElastiCache.
 - 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 des connexions dans Amazon CloudWatch à l'aide de `CurConnections` 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] Comprimez 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] :
 - [Amazon ElastiCache for Redis - Entrepôt de données mondial](#)
 - [Paramètres spécifiques Memcached](#)
 - [Amazon ElastiCache for Redis 5.0.3 améliore la gestion des E/S pour des performances boostées](#)
 - [Configure Amazon ElastiCache for Redis for higher availability](#)

EP 5 : Comment gérez-vous la suppression et/ou l'expulsion de clés ?

Introduction au niveau de la question : les charges de travail ont des exigences différentes et un comportement attendu lorsqu'un nœud de cluster approche des limites de consommation de mémoire. Amazon ElastiCache for Redis applique différentes politiques 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 de `maxmemory-policy` sur le groupe de paramètres du cluster ElastiCache for Redis.

- 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 cluster Amazon ElastiCache 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.
 - Pour les charges de travail ElastiCache for Redis, lorsque vous supprimez explicitement des clés du cluster, le comportement de la commande `UNLINK` équivaut à celui de `DEL` : elle supprime les 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 ElastiCache for Redis 6.x, le comportement de la commande `DEL` peut être modifié dans le groupe de paramètres à l'aide du paramètre `lazyfree-lazy-user-del`.
- [Ressources] :
 - [Configuration des paramètres de moteur à l'aide de groupes de paramètres](#)
 - [UNLINK](#)
 - [Gestion financière dans le cloud avec AWS](#)

EP 6 : Comment modélisez-vous et interagissez-vous avec les données dans ElastiCache ?

Introduction au niveau de la question : ElastiCache est fortement tributaire des applications concernant les structures de données et le modèle de données utilisés, mais il doit également prendre en compte l'entrepôt de données sous-jacent (le cas échéant). Découvrez les structures de données ElastiCache for Redis disponibles et assurez-vous d'utiliser les structures de données les mieux adaptées à vos besoins.

Avantage au niveau de la question : la modélisation des données dans ElastiCache comporte plusieurs niveaux, notamment le cas d'utilisation de l'application, les types de données et les relations

entre les éléments de données. De plus, chaque type de données et chaque commande ElastiCache for Redis possèdent 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](#)
- [Meilleure pratique] Les commandes ElastiCache for Redis ont une complexité temporelle définie par la notation Big O. La complexité temporelle d'une commande est une représentation algorithmique/mathématique 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 d'ElastiCache for Redis, un grand nombre d'opérations à complexité temporelle élevée se traduira par une baisse des performances et des délais d'exécution potentiels.

[Ressources] :

- [Commandes](#)
- [Meilleure pratique] Utilisez les API pour obtenir une meilleure visibilité de l'interface graphique sur le modèle de données de votre cluster.

[Ressources] :

- [Redis Commander](#)
- [Redis Browser](#)
- [Redsmin](#)

EP 7 : Comment enregistrez-vous les commandes s'exécutant lentement dans votre cluster Amazon ElastiCache ?

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 identifiant le temps nécessaire

à l'exécution des commandes, vous pouvez déterminer celles qui entraînent des performances médiocres et celles qui empêchent le moteur de fonctionner de manière optimale. Amazon ElastiCache for Redis permet également de transmettre ces informations à Amazon CloudWatch ou à 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] Amazon ElastiCache for Redis exécutant la version 6.0 ou ultérieures du moteur, un groupe de paramètres correctement configuré et la journalisation SLOWLOG activée sur le cluster.
 - Les paramètres requis ne sont disponibles que lorsque la compatibilité avec la version du moteur est définie sur Redis version 6.0 ou ultérieures.
 - 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.
- [Meilleure pratique] Tirez profit des fonctionnalités de CloudWatch ou de 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 envoyer des notifications d'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 des autorisations IAM pour publier sur CloudWatch ou 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 s'exécuter dans le moteur Redis avant d'être journalisée en tant que commande s'exécutant lentement. 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 s'exécutant lentement capturées dans la mémoire Redis 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 plus large à partir de `redis-cli` via la commande `SLOWLOG`, envisagez d'augmenter cette valeur. Cela permet de conserver davantage de commandes dans la mémoire Redis.

Si vous envoyez les données `SLOWLOG` à CloudWatch Logs ou à Kinesis Data Firehose, les données seront conservées et pourront être analysées en dehors du système ElastiCache, ce qui réduira la nécessité de stocker un grand nombre de commandes s'exécutant lentement dans la mémoire Redis.

- [Ressources] :
 - [How do I turn on Redis Slow log in an ElastiCache for Redis cache cluster?](#)
 - [Livraison de journaux](#)
 - [Paramètres spécifiques Redis](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

EP 8 : Comment l'autoscaling contribue-t-il à améliorer les performances du cluster ElastiCache ?

Introduction au niveau de la question : en implémentant la fonctionnalité d'autoscaling de Redis, vos composants ElastiCache peuvent s'ajuster au fil du temps pour augmenter ou diminuer automatiquement le nombre de partitions ou de réplicas souhaités. Pour ce faire, vous pouvez implémenter le suivi des cibles ou une politique de mise à l'échelle planifiée.

Avantage au niveau de la question : 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. La fonctionnalité d'autoscaling d'ElastiCache for Redis surveille en permanence votre utilisation de l'UC et de la mémoire pour s'assurer que votre cluster fonctionne aux niveaux de performance souhaités.

- [Obligatoire] Lors du lancement d'un cluster pour ElastiCache for Redis :

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 Redis \(Mode Cluster activé\)](#)
- [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 métriques CloudWatch correspondantes pour l'utilisation du cluster sur une période de quatre semaines afin de déterminer le seuil de la 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 métriques CloudWatch](#)
- [Pratique encore meilleure] Il est recommandé de tester votre application avec les charges de travail minimales et maximales attendues, afin d'identifier le nombre exact de partitions/réplicas requis pour que le cluster puisse développer des politiques de mise à l'échelle et atténuer les problèmes de disponibilité.

[Ressources] :

- [Enregistrement d'une cible évolutive](#)
- [Enregistrement d'une cible évolutive](#)

Cadre Amazon ElastiCache Well-Architected - Pilier Optimisation des coûts

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

- [POC 1 : Comment identifiez-vous et suivez-vous les coûts associés à vos ressources ElastiCache ? Comment développez-vous des mécanismes permettant aux utilisateurs de créer, gérer et supprimer les ressources créées ?](#)
- [POC 2 : Comment utilisez-vous les outils de surveillance continue pour vous aider à optimiser les coûts associés à vos ressources ElastiCache ?](#)
- [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 ?](#)

POC 1 : Comment identifiez-vous et suivez-vous les coûts associés à vos ressources ElastiCache ? 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. La mise en place de processus et d'outils permettant de suivre les ressources créées entre le développement, la production et le retrait vous permet de gérer les coûts associés à ElastiCache.

Avantage au niveau de la question : le suivi continu de tous les coûts associés à votre charge de travail nécessite une compréhension approfondie de l'architecture qui inclut ElastiCache comme 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] Fondez un centre d'excellence cloud (CCoE) doté de l'une de ses chartes fondatrices, qui définira, suivra et prendra des mesures concernant les métriques relatives à l'utilisation d'ElastiCache par votre organisation. Si un CCoE existe et fonctionne, assurez-vous qu'il sait comment lire et suivre les coûts associés à 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. Le CCoE doit identifier, définir et publier des métriques de coût qui sont régulièrement mises à jour (tous les mois) concernant l'utilisation principale d'ElastiCache sur des 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. Le CCoE se compose d'une équipe interfonctionnelle constituée de représentants non exclusifs des équipes d'ingénierie logicielle, gestion des données, produit, finances et direction de votre organisation.

[Ressources] :

- [Create a Cloud Center of Excellence](#)
- [Tarification Amazon ElastiCache](#)
- [Obligatoire] Utilisez des balises de répartition des coûts pour suivre les coûts avec un faible niveau de granularité. Utilisez AWS Cost Management pour visualiser, comprendre et gérer vos coûts AWS et votre utilisation au fil du temps.
 1. Utilisez des balises pour organiser vos ressources et des balises de répartition des coûts pour effectuer le suivi de vos coûts AWS à un niveau détaillé. Une fois les balises de répartition des coûts activées, AWS les utilise pour organiser les coûts de vos ressources dans votre rapport

de répartition des coûts, de sorte que vous puissiez plus facilement classer vos coûts AWS par catégorie et en effectuer le suivi. AWS fournit deux types de balises de répartition des coûts : les balises générées par AWS et les balises définies par l'utilisateur. AWS définit, crée et applique pour vous les balises générées par AWS, tandis que vous définissez, créez et appliquez les 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. Vous pouvez utiliser des balises répartition des coûts pour organiser votre facture AWS afin de refléter votre propre structure de coût. Lorsque vous ajoutez des balises de répartition des coûts à vos ressources dans Amazon ElastiCache, vous pouvez suivre les coûts en regroupant les dépenses sur vos factures par des valeurs de balise de ressource. Vous devriez également envisager de combiner des balises pour suivre les coûts plus détaillés.

[Ressources] :

- [Utilisation des balises de répartition des coûts AWS](#)
 - [Surveillance des coûts avec des balises de répartition des coûts](#)
 - [AWS Cost Explorer](#)
- [Meilleure pratique] Connectez le coût d'ElastiCache à des métriques 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 utilisateurs/semaine, durée de session/utilisateur, taux d'abandon, taux d'accès au cache et clés suivies

[Ressources] :

- [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)
- [Bonne pratique] Maintenez une visibilité architecturale et opérationnelle à jour sur les métriques et les coûts pour l'ensemble de la charge de travail qui utilise ElastiCache.
1. Comprenez l'ensemble de votre écosystème de solutions. ElastiCache a tendance à faire partie d'un écosystème complet de services AWS au sein de son ensemble technologique, des clients à API Gateway, Redshift et QuickSight pour les outils de création de rapports (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 augmentent et/ou vous aident à gérer le coût global.

3. Votre diagramme doit inclure l'utilisation du calcul, du réseau, du stockage, des politiques de cycle de vie, de la collecte de métriques ainsi que les éléments ElastiCache 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 puissiez disposer d'une stratégie de gestion des coûts efficace pour votre ElastiCache.

POC 2 : Comment utilisez-vous les outils de surveillance continue pour vous aider à optimiser les coûts associés à vos ressources ElastiCache ?

Introduction au niveau de la question : vous devez trouver un juste équilibre entre le coût de votre ElastiCache et les métriques de performance de l'application. Amazon CloudWatch vous permet d'avoir une certaine visibilité sur les métriques opérationnelles clés qui peuvent vous aider à déterminer si vos ressources ElastiCache 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 sur-alloué et vous devez être en mesure de développer des mécanismes appropriés pour redimensionner vos ressources ElastiCache tout en répondant à vos besoins opérationnels, de disponibilité, de résilience et de performances.

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 des ressources ElastiCache surdimensionnées pendant de longues périodes.

- [Obligatoire] Utilisez CloudWatch pour surveiller vos clusters ElastiCache et analyser la relation entre ces métriques et vos tableaux de bord AWS Cost Explorer.
 1. ElastiCache fournit deux métriques au niveau de l'hôte (par exemple, utilisation du CPU) et des métriques qui sont spécifiques au logiciel de moteur de cache (par exemple, réussites d'accès

au cache et échecs d'accès au cache). Ces métriques sont mesurées et publiées pour chaque nœud de cache toutes les 60 secondes.

2. Les métriques de performance d'ElastiCache (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections et Evictions) peuvent indiquer que vous devez augmenter/diminuer la capacité (utiliser des types de nœuds de cache plus ou moins grands) ou diminuer/monter en puissance (ajouter plus/moins de 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 métriques CloudWatch](#)
 - [Quelles métriques dois-je surveiller ?](#)
 - [Tarification Amazon ElastiCache](#)
- [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 faire fonctionner vos clusters ElastiCache à l'aide de nœuds à la demande jusqu'à ce que vous ayez collecté suffisamment de données pour estimer vos besoins en 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 correct est sélectionné, vous devez examiner régulièrement le temps d'inactivité de l'objet ElastiCache.

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. Disponibles uniquement pour les clusters ElastiCache for Redis.
 2. Seuls certains types d'instances prennent en charge la hiérarchisation des données.
 3. Seule la version 6.2 d'ElastiCache for Redis 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](#)
- [Tarification Amazon ElastiCache](#)
- [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 fréquemment utilisées et les données moins fréquemment utilisées 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. Il se peut que vous deviez faire fonctionner vos clusters ElastiCache à l'aide d'instances ne prenant pas en charge la hiérarchisation des données jusqu'à ce que vous compreniez mieux vos besoins.
 2. Analysez le modèle d'utilisation des données de vos clusters ElastiCache.
 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](#)
- [Tarification Amazon ElastiCache](#)

Étapes de dépannage courantes et meilleures pratiques

Rubriques

- [Problèmes de connexion](#)
- [Erreurs du client Redis](#)
- [Résolution des problèmes de latence élevée en mode ElastiCache Serverless](#)
- [Résolution des problèmes de régulation dans Serverless ElastiCache](#)
- [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. Pour en savoir plus sur la connexion à un cache compatible TLS, cliquez [ici](#).
2. VPC : les ElastiCache caches ne sont accessibles que depuis un VPC. Assurez-vous que l'instance EC2 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 réside votre instance EC2 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 instance EC2. 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 Redis. Pour en savoir plus sur la configuration de l'accès aux ports, [cliquez ici](#).

Erreurs du client Redis

ElastiCache Le mode Serverless est uniquement accessible via les clients Redis qui prennent en charge le protocole en mode cluster Redis. Les clusters conçus par nos soins sont accessibles à partir des clients Redis dans l'un ou l'autre mode, en fonction de la configuration du cluster.

Si vous rencontrez des erreurs Redis dans votre client, tenez compte des points suivants :

1. Mode cluster : Si vous rencontrez des erreurs CROSSLOT ou des erreurs avec la commande [SELECT](#) Redis, vous essayez peut-être d'accéder à un cache activé en mode cluster avec un client Redis qui ne prend pas en charge le protocole Redis Cluster. ElastiCache Serverless ne prend en charge que les clients Redis qui prennent en charge le protocole de cluster Redis. Si vous souhaitez utiliser Redis 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 relatives à la configuration des clients Redis, 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 du côté client, mais aucune augmentation correspondante, `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 AWS 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 Redis 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 dans les mêmes zones de disponibilité que votre cache : vous pouvez observer une latence plus élevée côté client si votre application n'est pas déployée dans les mêmes zones de disponibilité 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 les mêmes AZ. 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 Redis 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 Redis, 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 demandes : dans ElastiCache Redis, 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 ECPU/seconde (90 000 ECPU/seconde lors de l'utilisation de Read from Replica) sur un seul emplacement, dans une charge de travail qui exécute de simples commandes 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 Redis intègrent des nouvelles tentatives 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 en fonction de votre charge de travail. ElastiCache Le mode Serverless permet généralement d'augmenter rapidement la taille de stockage, en doublant la taille de stockage de votre cache en 10 à 12 minutes.
- **Distribution uniforme des clés et des demandes** : dans ElastiCache Redis, 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 adressées à un seul emplacement dépasse 30 000 ECPUS/seconde, dans une charge de travail qui exécute de simples commandes SET/GET.
- **Lire depuis une réplique** : si votre application le permet, pensez à utiliser la fonction « Lire depuis une réplique ». La plupart des clients Redis 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 90 000 ECPU/seconde sur un seul emplacement, pour les charges de travail utilisant de simples commandes SET/GET.

Rubriques connexes

- [Étapes de dépannage supplémentaires](#)
- [the section called “Bonnes pratiques et stratégies de mise en cache”](#)

Étapes de dépannage supplémentaires

Les éléments suivants doivent être vérifiés lors de la résolution des problèmes de connectivité persistants avec ElastiCache :

Rubriques

- [Groupes de sécurité](#)
- [Listes ACL réseau](#)
- [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](#)
- [Dépannage 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 (instance EC2, AWS Lambda fonction, conteneur Amazon ECS, etc.) et ElastiCache votre 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 vers. ElastiCache Le port par défaut est 11211 pour Memcached

et 6379 pour Redis. 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](#).

Listes ACL réseau

Les listes de contrôle d'accès réseau (ACL) sont des règles sans état. Le trafic doit être autorisé dans les deux sens (entrant et sortant) pour réussir. Les listes d'accès réseau sont affectées à des sous-réseaux et non à des ressources spécifiques. Il est possible que la même ACL soit assignée à ElastiCache et à la même ressource client, en particulier s'ils se trouvent dans le même sous-réseau.

Par défaut, les listes ACL réseau autorisent 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 Redis 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 d'adresses IP spécifiques 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 d'adresses IP spécifiques peut créer des problèmes en cas de basculement ou de dimensionnement du cluster)
 - Autoriser/Refuser : Autoriser

Pour plus d'informations, veuillez consulter [ACL réseau](#).

Tables de routage

De même que les ACL réseau, 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.

Des environnements plus complexes, impliquant plusieurs VPC, un routage dynamique ou des pare-feu réseau, peuvent devenir 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 ne pas utiliser le service DNS VPC) peuvent ne pas connaître 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) `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** : Redis est une application 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 Redis et `CPUUtilization` l'utilisation de tous les vCPU. Les nœuds avec plus d'un vCPU ont généralement des valeurs différentes pour `CPUUtilization` et `EngineCPUUtilization`, le second étant généralement plus élevé. 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 réplica, d'activer les opérations de lecture via la commande Redis [READONLY](#) et de soumettre les demandes de lecture aux réplicas.
- Nombre élevé de connexions :
 - `CurrConnections` et `NewConnections` : `CurrConnection` est le nombre de connexions établies au moment de la collecte 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. Redis offre 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 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 plus d'informations, consultez [Limites liées au réseau](#).

- Commandes complexes : les commandes Redis sont servies sur un seul thread, ce qui signifie que les demandes sont servies de façon 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 keypace entier en recherchant un modèle donné et bloque l'exécution d'autres commandes lors de son exécution. Redis 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 keypace 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 keypace 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 keypace.

En outre KEYS, plusieurs autres commandes sont potentiellement nuisibles si elles ne sont pas utilisées correctement. Pour afficher la liste de toutes les commandes et leur complexité temporelle respective, accédez à <https://redis.io/commands>.

Exemples de problèmes potentiels :

- **Scripts Lua** : Redis fournit un interpréteur Lua intégré, permettant l'exécution de scripts côté serveur. Les scripts Lua sur Redis sont exécutés au niveau du moteur et sont atomiques par définition, ce qui signifie qu'aucune autre commande ou aucun autre 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, et autres directement sur le moteur Redis. 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 keypace 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 de plusieurs modifications de clés associées dans Redis. 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 est UNLINK, qui est une commande asynchrone disponible depuis Redis 4. UNLINK doit avoir la préférence sur DEL dès que possible. À partir ElastiCache de Redis 6x, le `lazy-free-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 rubrique [Modifications des paramètres Redis 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 fournit beaucoup plus de 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 fournit également des commandes qui agissent sur une ou plusieurs clés, quel que soit leur type de données. ElastiCache for Redis 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, veuillez consulter <https://redis.io/commands>.

Validation de connectivité réseau

Après avoir examiné les configurations réseau liées à la résolution DNS, aux groupes de sécurité, aux listes ACL réseau et aux tables de routage, la connectivité peut être validée à l'aide de VPC Reachability Analyzer 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. Accédez à <https://console.aws.amazon.com/ec2/v2/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. Créez un chemin d'analyse sur <https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer> et choisissez les options suivantes :
 - Type de source : Choisissez une instance si votre ElastiCache client fonctionne sur une instance Amazon EC2 ou une interface réseau (s'il utilise un autre service, tel qu' AWS Fargate Amazon ECS avec un réseau awsvpc, etc.) AWS Lambda, et l'ID de ressource correspondant (instance EC2 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 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 possède pas les bons indicateurs de configuration ou les 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 `CurConnections` 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 de trafic réseau : vérifiez les [CloudWatch indicateurs suivants pour Redis](#) afin d'identifier les éventuelles limites du réseau atteintes 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. Les instances Amazon EC2 ont également des limites réseau qui peuvent être suivies via des [métriques de pilote 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 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.

- Mesures de latence : ElastiCache for Redis fournit des CloudWatch métriques permettant de 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 Redis :
 - info commandstats : fournit une liste des commandes exécutées depuis le démarrage du moteur Redis, leur nombre d'exécutions cumulées, leur temps d'exécution total et leur 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 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 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 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 `CurConnections`. 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 n'envoie aucune demande au cours de la période, le moteur Redis résilie la connexion dès que la connexion atteint la valeur de 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 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 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 client habituels, les options suivantes contrôlent le tampon pour les nœuds de réplica 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`;

Dépannage côté client pour les instances Amazon EC2

La charge et la réactivité du côté client peuvent également affecter les demandes adressées à ElastiCache. Les limites des instances EC2 et du système d'exploitation doivent être examinées attentivement lors du dépannage des problèmes intermittents de connectivité ou de délai d'expiration. Quelques points clés à observer :

- UC :
 - Utilisation de l'UC de l'instance EC2 : assurez-vous que l'UC n'a pas été saturée ou proche de 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 [instances EC2 à capacité extensible](#), assurez-vous que leur solde de crédit d'UC 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 la rubrique [Types d'instance Amazon EC2](#).
 - 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 se compose de milliers ou de millions de très petites demandes 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 commande PING Redis avec 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"], 0x7ffffede7cc38 /* 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\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\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.

Sécurité dans Amazon ElastiCache

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

La sécurité est une responsabilité partagée entre AWS et vous-même. Le [modèle de responsabilité partagée](#) décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

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

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Amazon ElastiCache. Les rubriques suivantes vous montrent comment configurer Amazon ElastiCache pour répondre à vos objectifs de sécurité et de conformité. Vous pouvez également apprendre à utiliser d'autres services AWS qui vous aident à contrôler et sécuriser vos ressources Amazon ElastiCache.

Rubriques

- [Protection des données dans Amazon ElastiCache](#)
- [Confidentialité du trafic inter-réseau](#)
- [Identity and Access Management pour Amazon ElastiCache](#)
- [Validation de conformité pour Amazon ElastiCache](#)
- [Résilience dans Amazon ElastiCache](#)
- [Sécurité de l'infrastructure dans AWS ElastiCache](#)
- [Mises à jour du service dans ElastiCache](#)

Protection des données dans Amazon ElastiCache

Le [modèle de responsabilité partagée](#) AWS 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 globale sur laquelle l'ensemble du cloud AWS s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure est de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité des services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, veuillez consulter la [FAQ sur la confidentialité des données](#).

À des fins de protection des données, nous vous recommandons de protéger les autorisations du compte AWS et de configurer les 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 TLS pour communiquer avec des ressources AWS.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données 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 s'applique aussi lorsque vous utilisez ElastiCache ou d'autres services AWS à l'aide de la console, de l'API, d'AWS CLI ou des kits AWS SDK. Toutes les données que vous entrez dans ElastiCache ou d'autres services peuvent être récupérées afin d'être insérées dans des journaux de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Rubriques

- [Sécurité des données dans Amazon ElastiCache](#)

Sécurité des données dans Amazon ElastiCache

Afin de vous aider à préserver la sécurité de vos données, Amazon ElastiCache et Amazon EC2 fournissent les mécanismes permettant de protéger vos données sur le serveur de tout accès non autorisé.

Amazon ElastiCache for Memcached fournit également des fonctionnalités de chiffrement pour les données des caches exécutés sur Memcached versions 1.6.12 ou versions ultérieures.

- 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.

Rubriques

- [ElastiCache chiffrement en transit \(TLS\)](#)
- [Chiffrement au repos dans ElastiCache](#)

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 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 du chiffrement en transit](#)
- [Bonnes pratiques en matière de chiffrement en transit](#)

- [Activation du chiffrement en transit](#)
- [Connexion aux nœuds activés avec le chiffrement en transit à l'aide d'openssl](#)
- [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.

Conditions du chiffrement en transit

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 plus d'informations, consultez [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. Référez 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.

Activation du chiffrement en transit

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, voir [Création d'un cluster Memcached \(console\)](#).

Connexion aux nœuds activés avec le chiffrement en transit à l'aide d'openssl

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 contexte SSLContext approprié :

```
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 contexte SSLContext approprié :

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery 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/mem-ug/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 cluster ElastiCache pour .NET](#).

Chiffrement au repos dans ElastiCache

Afin de vous aider à préserver la sécurité de vos données, Amazon ElastiCache et Amazon S3 fournissent différentes façons de restreindre l'accès à vos données lorsqu'elles se trouvent dans votre cache. Pour plus d'informations, consultez [Sécurité des VPC Amazon et ElastiCache](#) et [Identity and Access Management pour Amazon ElastiCache](#).

- Le disque lors des opérations de synchronisation et d'échange

ElastiCache propose un chiffrement au repos par défaut (géré par le service) ainsi que la possibilité d'utiliser vos propres clés KMS AWS symétriques gérées par le client dans [AWS Key Management Service \(KMS\)](#). 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](#).

Note

Le chiffrement par défaut (géré par le service) est la seule option disponible dans les régions GovCloud (US).

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 des clés gérées par le client depuis AWS KMS](#)
- [Activation du chiffrement au repos](#)
- [consultez aussi](#)

Conditions du chiffrement au repos

Il convient de garder à l'esprit les contraintes suivantes concernant le chiffrement au repos ElastiCache lorsque vous planifiez votre implémentation du chiffrement au repos ElastiCache :

- Le chiffrement au repos est uniquement pris en charge sur les caches sans serveur.
- L'option d'utiliser la clé gérée par le client pour le chiffrement au repos n'est pas disponible dans les régions AWS GovCloud (us-gov-east-1 et us-gov-west-1).

Utilisation des clés gérées par le client depuis 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, possédez et gérez dans votre compte AWS. 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 plus d'informations sur la façon de créer des clés racines AWS KMS, veuillez consulter [Création de clés](#) dans le Guide du développeur AWS Key Management Service.

ElastiCache permet l'intégration avec AWS KMS. Pour plus d'informations, veuillez consulter [Utilisation d'octrois](#) dans le Guide du développeur AWS Key Management Service. Aucune action n'est nécessaire de la part du client pour permettre l'intégration d'Amazon ElastiCache avec AWS KMS.

La clé de condition `kms:ViaService` limite l'utilisation d'une clé AWS KMS (clé KMS) aux demandes provenant de services AWS spécifiés. Pour utiliser `kms:ViaService` avec ElastiCache, incluez les deux noms `ViaService` dans la valeur de clé de condition : `elasticache.AWS_region.amazonaws.com` et `dax.AWS_region.amazonaws.com`. Pour de plus amples informations, veuillez consulter [kms:ViaService](#).

Vous pouvez utiliser [AWS CloudTrail](#) pour suivre les demandes qu'Amazon ElastiCache envoie à AWS Key Management Service en votre nom. Tous les appels d'API vers AWS Key Management Service liés aux clés gérées par le client ont des journaux CloudTrail correspondants. Vous pouvez également voir les accords qu'ElastiCache crée en appelant l'API KMS [ListGrants](#).

- 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 récupéré après un échec du matériel. 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 et n'a donc aucun impact sur votre capacité à accéder à vos données ElastiCache. Les caches Amazon

ElastiCache chiffrés ne prennent pas en charge la rotation manuelle des clés, laquelle implique de créer une nouvelle clé racine et de mettre à jour toute référence à l'ancienne clé. Pour en savoir plus, veuillez consulter [Rotation des clés AWS KMS](#) dans le Guide du développeur AWS Key Management Service.

- Le chiffrement d'un cache ElastiCache à l'aide d'une clé KMS exige un octroi par cache. Cet octroi est utilisé pendant toute la durée de vie du cache.
- Pour en savoir plus sur les octrois et les limites d'AWS KMS, veuillez consulter [Limites](#) dans le Guide du développeur AWS Key Management Service.

Activation du chiffrement au repos

Le chiffrement au repos est activé pour tous les caches sans serveur.

Vous pouvez activer le chiffrement au repos lorsque vous créez un cache ElastiCache. Vous pouvez effectuer cette opération à l'aide de la AWS Management Console, de la AWS CLI ou de l'API ElastiCache.

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 l'ID/ARN de clé d'AWS KMS pour le chiffrement au repos.

Pour plus d'informations sur la façon de créer des clés racines AWS KMS, veuillez consulter [Création de clés](#) dans le Guide du développeur AWS Key Management Service

Table des matières

- [Activation du chiffrement au repos à l'aide de l'AWS Management Console](#)

Activation du chiffrement au repos à l'aide de l'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 détenue par AWS est utilisée pour chiffrer les données. Pour choisir votre propre clé AWS KMS, 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 Clé AWS KMS.

consultez aussi

- [Sécurité des VPC Amazon et ElastiCache](#)
- [Identity and Access Management pour Amazon ElastiCache](#)

Confidentialité du trafic inter-réseau

Amazon ElastiCache utilise les techniques suivantes pour sécuriser vos données de cache et les protéger contre un accès non autorisé :

- [Sécurité des VPC Amazon et ElastiCache](#) 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.

Sécurité des VPC Amazon et ElastiCache

Étant donné que la sécurité des données est importante, ElastiCache vous fournit des moyens de contrôler qui a accès à vos données. La façon dont vous contrôlez l'accès à vos données dépend du fait d'avoir lancé ou non vos clusters dans un Virtual Private Cloud (Amazon VPC) ou Amazon EC2-Classic.

Important

Nous avons rendu obsolète l'utilisation de Amazon EC2-Classic pour le lancement des 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 cluster ElastiCache dans un Amazon VPC. Ces informations sont destinées aux utilisateurs qui souhaitent mieux comprendre comment ElastiCache et Amazon VPC fonctionnent ensemble.

Rubriques

- [Présentation d'ElastiCache et d'Amazon VPC](#)
- [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](#)

Présentation d'ElastiCache et d'Amazon VPC

ElastiCache est entièrement intégré au Amazon Virtual Private Cloud (Amazon VPC). Pour les utilisateurs ElastiCache, cela signifie que :

- Si votre compte AWS prend uniquement en charge la plateforme EC2-VPC, ElastiCache lance toujours votre cluster dans un Amazon VPC.
- Si vous découvrez AWS, 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 ?](#).

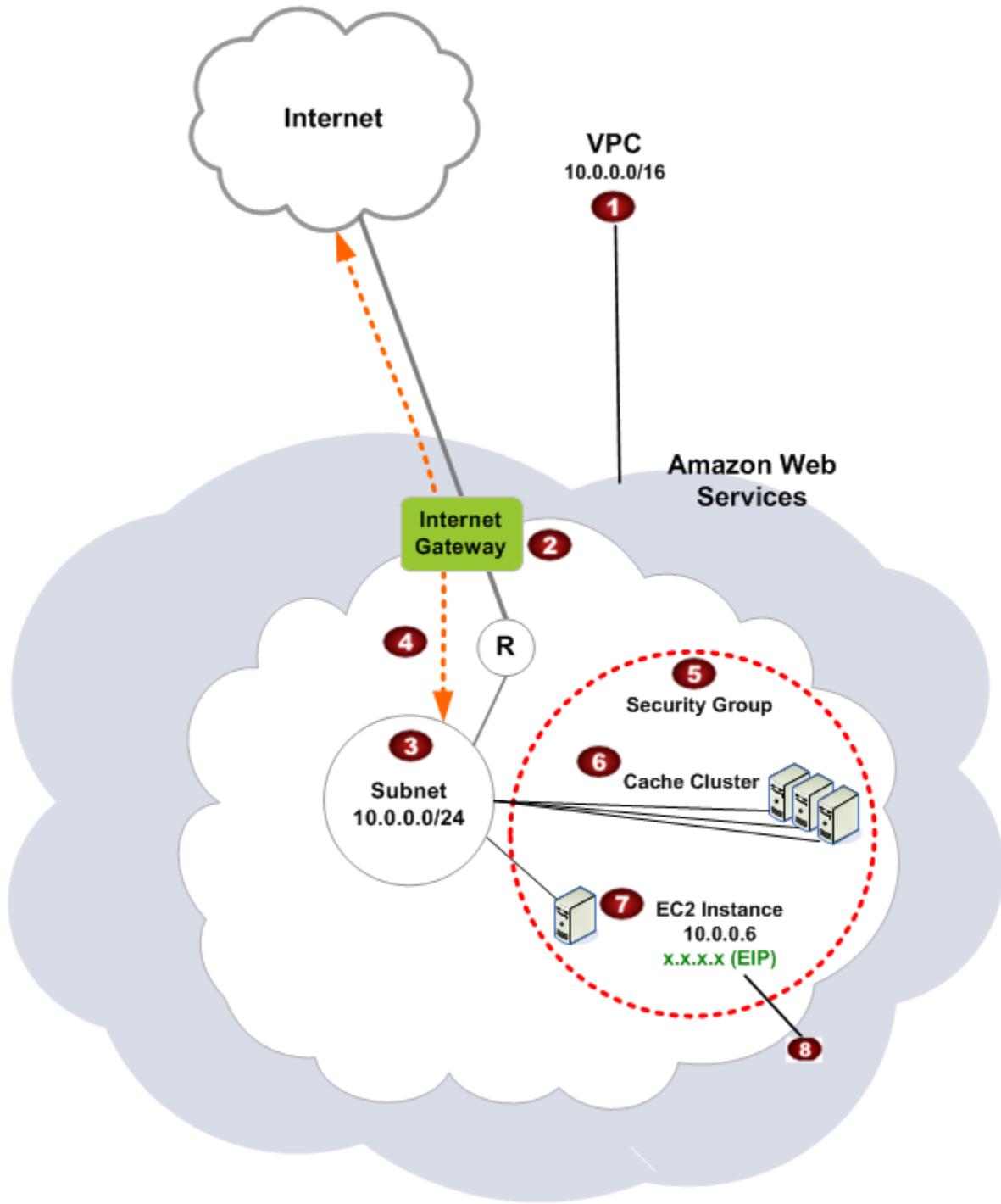
Amazon Virtual Private Cloud vous permet de créer un réseau virtuel dans le Cloud AWS qui ressemble beaucoup à un centre de données traditionnel. Vous pouvez configurer votre Amazon VPC en sélectionnant sa plage d'adresses IP, en créant des sous-réseaux et en configurant des tables de routage, des passerelles de réseau et des paramètres de sécurité.

Les fonctionnalités de base d'ElastiCache sont les mêmes que dans un cloud privé virtuel. ElastiCache gère les mises à niveau logicielles, les correctifs, la détection et la résolution des pannes que vos clusters soient déployés à l'intérieur ou à l'extérieur d'un Amazon VPC.

Les nœuds de cache ElastiCache déployés à l'extérieur d'un Amazon VPC reçoivent une adresse IP vers laquelle pointe le point de terminaison/nom DNS. Cela fournit une connectivité à partir des instances Amazon Elastic Compute Cloud (Amazon EC2). Lorsque vous lancez un cluster ElastiCache dans un sous-réseau privé Amazon VPC, chaque nœud de cache se voit attribuer une adresse IP privée dans ce sous-réseau.

Présentation d'ElastiCache dans un Amazon VPC

Le graphique et le tableau suivants décrivent l'environnement Amazon VPC avec des clusters ElastiCache et des instances Amazon EC2 lancées dans l'Amazon VPC.



1

L'Amazon VPC est une partie isolée du Cloud AWS qui se voit attribuer son propre bloc d'adresses IP.

2

Une passerelle Internet connecte votre Amazon VPC directement à Internet et donne accès à d'autres ressources AWS, telles qu'Amazon Simple Storage Service (Amazon S3) s'exécutant 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 où vous pouvez isoler les ressources AWS selon vos besoins en matière de sécurité et de fonctionnement.

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 de vos clusters ElastiCache et les instances Amazon EC2.

6

Vous pouvez lancer un cluster ElastiCache 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 instances Amazon EC2 dans le sous-réseau. Chaque instance Amazon EC2 dispose d'une adresse IP privée provenant de la plage d'adresses du sous-réseau. L'instance Amazon EC2 peut se connecter à un nœud de cache dans le même sous-réseau.

8

Pour qu'une instance Amazon EC2 dans votre Amazon VPC soit accessible depuis Internet, vous devez affecter une adresse statique, publique appelée « adresse IP élastique » à l'instance concernée.

Prérequis

Pour créer un cluster ElastiCache au sein d'un Amazon VPC, votre Amazon VPC doit répondre aux critères suivants :

- L'Amazon VPC doit autoriser les instances Amazon EC2 non dédiées. Vous ne pouvez pas utiliser ElastiCache dans un Amazon VPC qui est configuré pour la location de l'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 celui-ci à associer à vos points de terminaison de VPC ou à vos nœuds de cache.
- Les blocs d'adresses CIDR pour chaque sous-réseau doivent être suffisamment grands pour fournir des adresses IP inutilisées pour ElastiCache à utiliser lors des opérations 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 ressources AWS qui ne sont pas en cours d'exécution 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 devrez payer de frais supplémentaires liés à l'utilisation de la bande passante lorsque vous accédez aux ressources AWS via Internet.

Vous pouvez utiliser des groupes de sécurité Amazon VPC pour sécuriser les clusters ElastiCache et les instances Amazon EC2 dans votre Amazon VPC. Les groupes de sécurité agissent comme un pare-feu au niveau de l'instance, et non au niveau du sous-réseau.

Note

Nous vous recommandons 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
Utilisation d'Amazon VPC via la AWS Management Console	Amazon VPC User Guide

Description	Documentation
Description complète de toutes les commandes Amazon VPC	Référence des commandes en ligne Amazon EC2 (les commandes Amazon VPC se trouvent dans la référence Amazon EC2)
Descriptions complètes des opérations de l'API Amazon VPC, des types de données et des erreurs	Référence de l'API Amazon EC2 (les opérations d'API Amazon VPC se trouvent dans la référence Amazon EC2)
Informations pour l'administrateur de réseau qui a besoin de configurer la passerelle de votre côté avec une connexion VPN IPsec facultative	Description de AWS Site-to-Site VPN

Pour plus d'informations sur Amazon Virtual Private Cloud, veuillez consulter [Amazon Virtual Private Cloud](#).

Modèles d'accès pour accéder à un ElastiCache cache dans un Amazon VPC

Amazon ElastiCache prend en charge les scénarios suivants pour accéder à un cache dans un Amazon VPC :

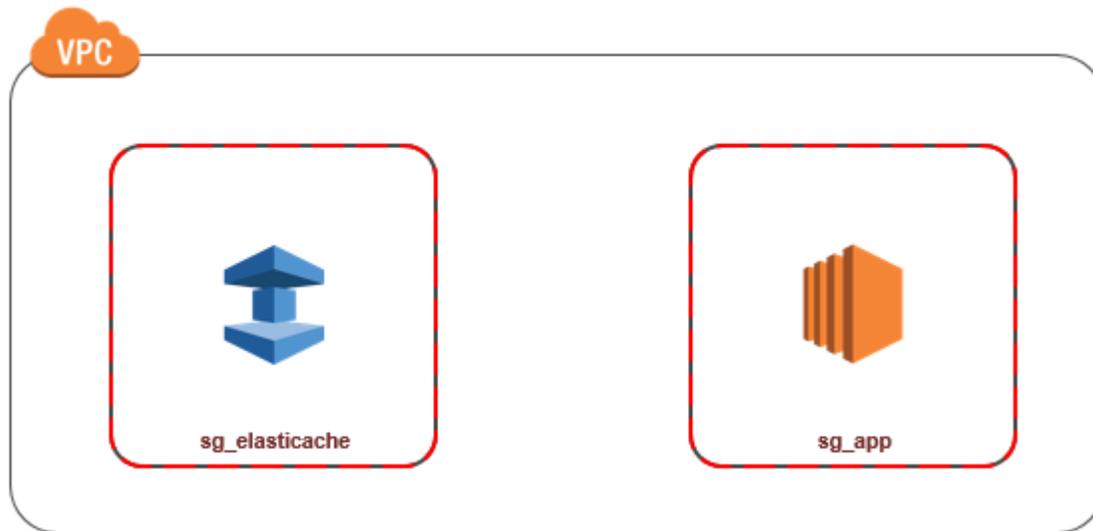
Table des matières

- [Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans le même Amazon VPC](#)
- [Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents](#)
 - [Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans la même région](#)
 - [Utilisation de Transit Gateway](#)
 - [Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans différentes régions](#)
 - [Utilisation de VPC en transit](#)
- [Accès à un ElastiCache cache à partir d'une application exécutée dans le centre de données d'un client](#)
 - [Accès à un ElastiCache cache à partir d'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'instance Amazon EC2 se trouvent dans le même Amazon VPC

Le cas d'utilisation le plus courant est une application déployée sur une instance EC2 qui doit se connecter à un cache dans le même VPC.

Le schéma suivant illustre ce scénario.



La solution la plus simple pour gérer l'accès entre les instances EC2 et les caches du même VPC consiste à 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.

2. Créez un groupe de sécurité VPC pour vos instances EC2 (serveurs web et d'application). Ce groupe de sécurité peut, si nécessaire, autoriser l'accès à l'instance EC2 à partir d'Internet via la table de routage du VPC. Par exemple, vous pouvez définir des règles sur ce groupe de sécurité pour autoriser l'accès TCP à l'instance EC2 sur le port 22.
3. Créez des règles personnalisées dans le groupe de sécurité de votre cache, qui autorisent les connexions depuis le groupe de sécurité que vous avez créé pour vos instances EC2. 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 l'adresse https://console.aws.amazon.com/vpc.](https://console.aws.amazon.com/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.
 - b. Dans le champ Source, saisissez l'ID de votre groupe de sécurité. Dans la liste, sélectionnez le groupe de sécurité que vous utiliserez pour vos instances Amazon EC2.
5. Choisissez Enregistrer lorsque vous avez terminé.

The screenshot shows the 'Edit inbound rules' dialog in the AWS console. It has a title bar with 'Edit inbound rules' and a close button. Below the title bar are four columns: 'Type', 'Protocol', 'Port Range', and 'Source'. The 'Type' dropdown is set to 'Custom TCP Rule'. The 'Protocol' field contains 'TCP'. The 'Port Range' field contains '6379'. The 'Source' dropdown is set to 'Custom', and a text input field next to it contains 'sg_app'. A dropdown menu is open below the 'Source' field, showing a list of security groups, with 'sg-99fc5ce6 - sg_app' selected. At the bottom left is an 'Add Rule' button. At the bottom right are 'Cancel' and 'Save' buttons.

Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des Amazon VPC différents

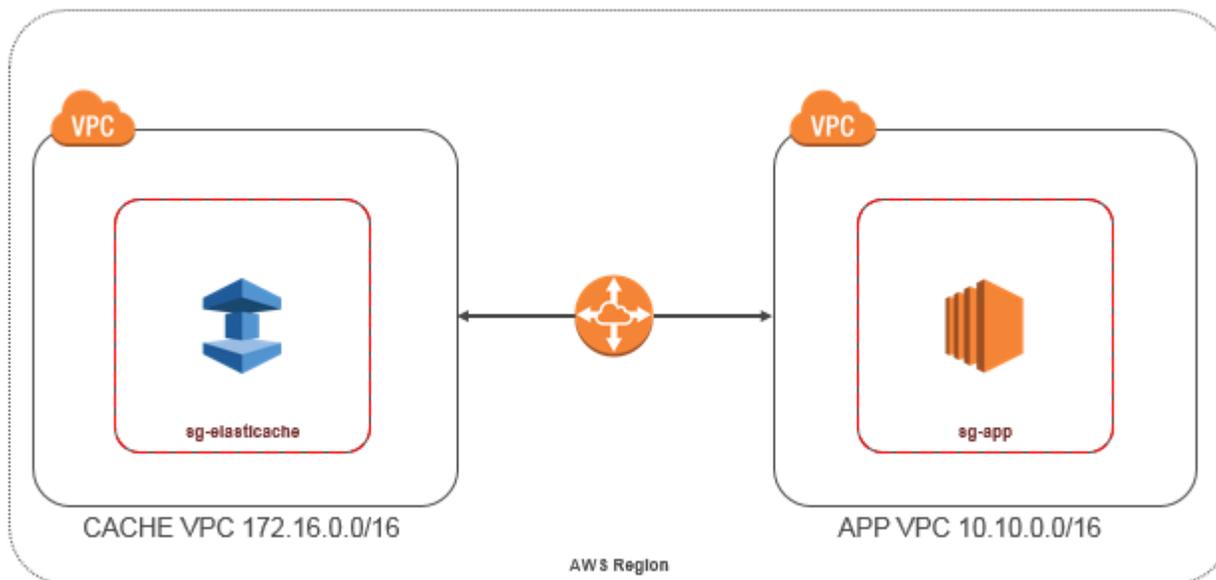
Quand votre cache se trouve dans un VPC différent de l'instance EC2 que vous utilisez pour y accéder, il existe plusieurs moyens d'accéder au cache. Si le cache et l'instance EC2 se trouvent dans des VPC différents, mais dans la même région, vous pouvez utiliser l'appairage de VPC. Si le cache et l'instance EC2 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'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans la même région](#)
- [Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans différentes régions](#)

Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans la même région

Le schéma suivant illustre l'accès à un cache par une instance Amazon EC2 située dans un autre réseau Amazon VPC de la même région à l'aide d'une connexion d'appairage de réseau Amazon VPC.



Cache auquel accède une instance Amazon EC2 située dans un autre réseau Amazon VPC de la même région : connexion d'appairage de VPC

Une connexion d'appairage de VPC est une connexion de mise en réseau entre deux VPC qui permet de router le trafic entre ces derniers à l'aide d'adresses IP privées. Les instances des deux VPC peuvent communiquer entre elles comme si elles se trouvaient dans le même réseau. Vous pouvez créer une connexion d'appairage VPC entre vos propres Amazon VPC ou avec un Amazon VPC d'un autre AWS compte au sein d'une même région. Pour en savoir plus sur l'appairage d'Amazon VPC, veuillez consulter la [documentation VPC](#).

Note

La résolution des noms DNS peut échouer pour les VPC homologues, selon les configurations appliquées au ElastiCache VPC. Pour résoudre ce problème, les deux VPC doivent être activés pour les noms d'hôtes 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. Veillez à ce que les plages IP des deux VPC ne se chevauchent pas ou vous ne pourrez pas les appairer.
2. Appairez les deux VPC. Pour de plus amples informations, veuillez consulter [Création et acceptation d'une connexion d'appairage d'Amazon VPC](#).
3. Mettez à jour votre table de routage. Pour de plus amples informations, veuillez consulter [Mise à jour de vos tables de routage pour une connexion d'appairage de VPC](#).

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	Destination	Target
172.16.0.0/16	local	10.10.0.0/16	local
10.10.0.0/16	pcx-a894f1c1	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 des VPC et des connexions VPN dans la même AWS région et d'acheminer le trafic entre eux. Une passerelle de transit fonctionne sur plusieurs AWS comptes, et vous pouvez utiliser AWS Resource Access Manager pour partager votre passerelle de transit avec d'autres comptes. Une fois que vous avez partagé une passerelle de transit avec un autre AWS compte, le propriétaire du compte peut associer ses VPC à votre passerelle de transit. Un utilisateur de l'un des comptes peut supprimer l'attachement à tout moment.

Vous pouvez activer la multicast sur une passerelle de transit, puis créer un domaine multicast de passerelle de transit qui autorise l'envoi du trafic multicast à partir de votre source multicast vers des membres de groupe multicast sur des attachements de VPC que vous associez au domaine.

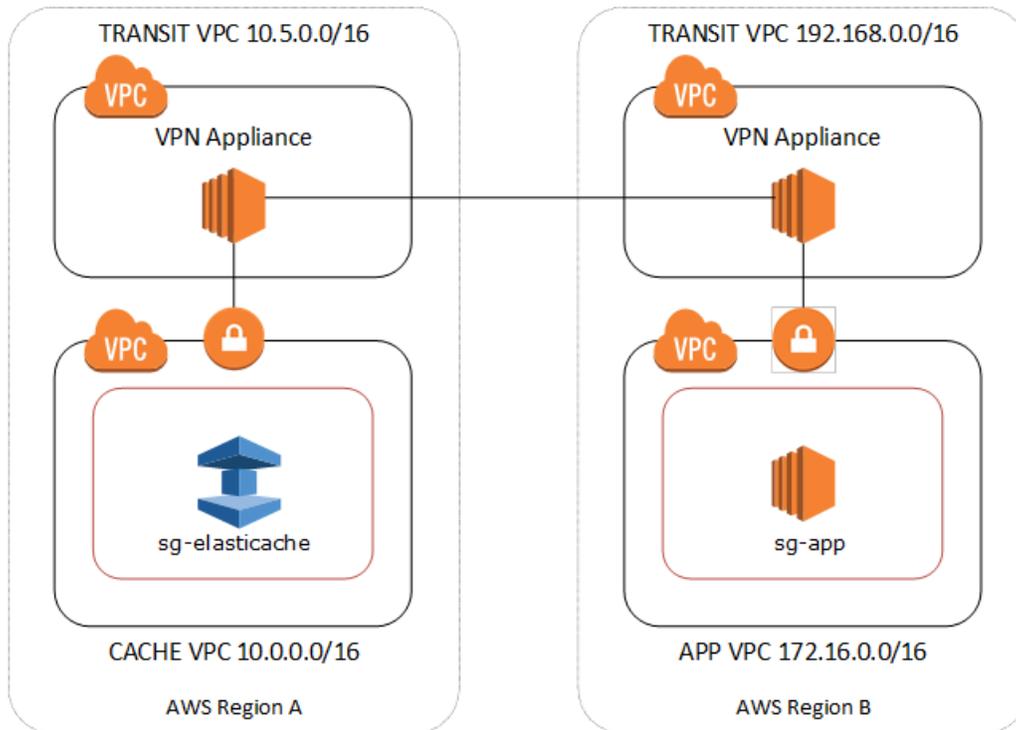
Vous pouvez également créer une pièce jointe de connexion d'appairage entre les passerelles de transport en commun de différentes AWS régions. Cela vous permet d'acheminer le trafic entre les attachements des passerelles de transit dans différentes régions.

Pour plus d'informations, consultez [Passerelles de transit](#).

Accès à un ElastiCache cache lorsque celui-ci et l'instance Amazon EC2 se trouvent dans des VPC Amazon différents dans différentes régions

Utilisation de VPC en transit

Une alternative à l'utilisation de l'appairage de VPC, une autre stratégie courante pour connecter plusieurs VPC situés dans différentes zones géographiques et réseaux distants consiste à créer un VPC en transit faisant office de centre de transit dans le réseau mondial. Un VPC en transit simplifie la gestion du réseau et limite le nombre de connexions requises pour connecter plusieurs VPC et réseaux distants. Cette structure permet de gagner du temps et de l'énergie, ainsi que de réduire les coûts. Elle est en effet implémentée virtuellement et évite donc les dépenses traditionnelles liées à l'implantation physique dans un hub de transit de colocalisation ou au déploiement d'un matériel réseau physique.



Connexion sur différents VPC de différentes régions

Une fois le VPC Transit Amazon établi, une application déployée dans un VPC « relais » d'une région peut se connecter à un 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 mettez en cache les VPC pour acheminer le trafic via la passerelle réseau privé virtuel 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 à partir d'une application exécutée dans le centre de données d'un client

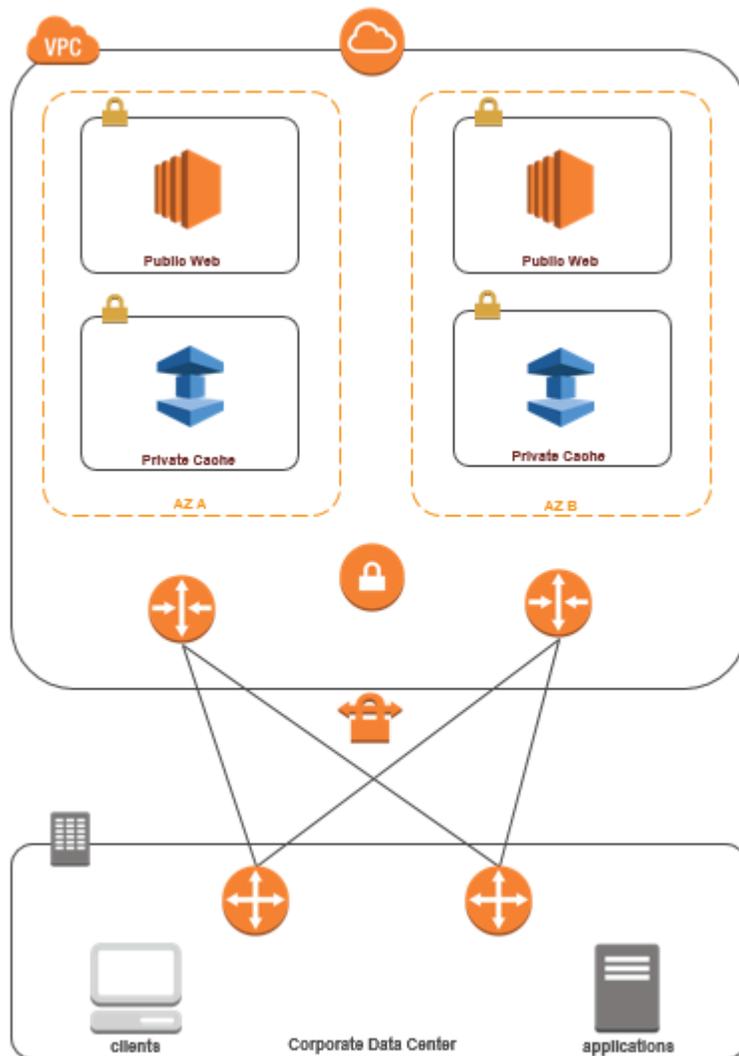
Un autre scénario possible est une architecture hybride dans laquelle les clients ou les applications du centre de données du client peuvent avoir besoin d'accéder à un ElastiCache cache dans le VPC. Ce scénario est également pris en charge, à condition qu'une connectivité existe entre le VPC d'un client et le centre de données, via un VPN ou via Direct Connect.

Rubriques

- [Accès à un ElastiCache cache à partir d'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 à partir d'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

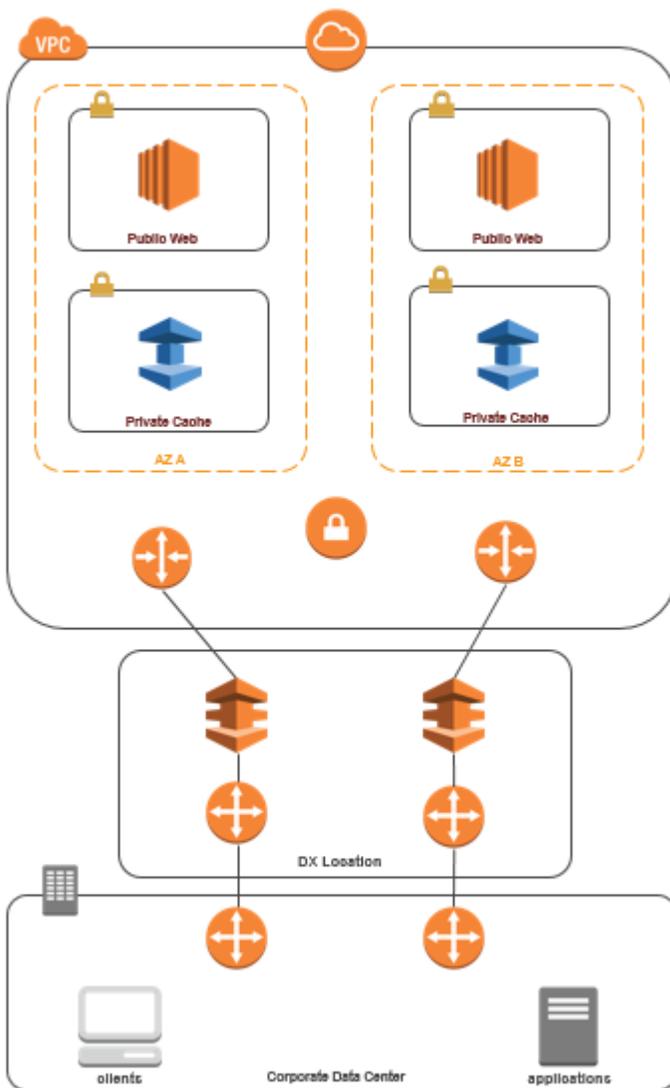
Pour accéder à un cache dans un VPC à partir d'une application sur site via une connexion VPN

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 à partir d'une application exécutée sur votre réseau à l'aide de Direct Connect

1. Etablissez une connectivité Direct Connect. Pour plus d'informations, voir [Getting Started with AWS Direct Connect](#).
2. Modifiez le groupe de sécurité de votre 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.

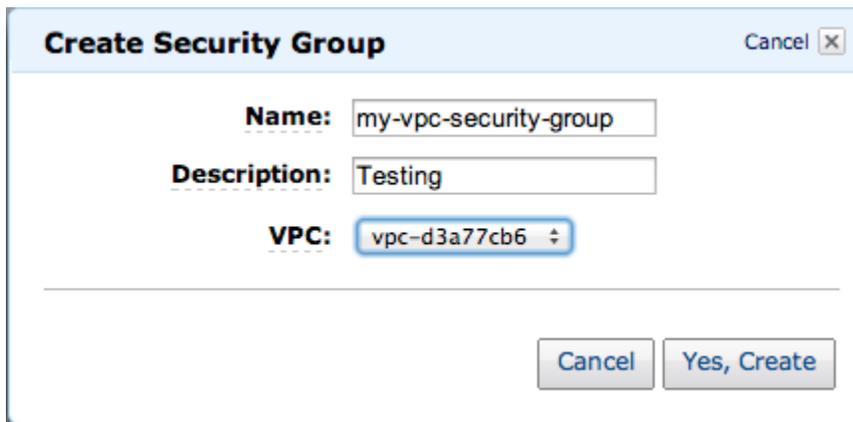
Création d'un Amazon VPC (console)

1. Connectez-vous à la Console de gestion AWS et ouvrez la console Amazon VPC à l'adresse <https://console.aws.amazon.com/vpc/>.
2. Dans le tableau de bord VPC, choisissez Create VPC (Créer un VPC).
3. Sous Resources to create (Ressources à créer), choisissez VPC and more (VPC et autres).
4. Sous Number of Availability Zones (AZs) (Nombre de zones de disponibilité (AZ)), choisissez le nombre de zones de disponibilité dans lesquelles vous voulez lancer vos sous-réseaux.
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 lorsque vous lancerez vos clusters et ajouterez une instance Amazon EC2 à votre Amazon VPC.

7. Créez un groupe de sécurité Amazon VPC. Vous allez utiliser ce groupe pour votre cluster de cache et votre instance Amazon EC2.
 - 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.



Create Security Group Cancel

Name: my-vpc-security-group

Description: Testing

VPC: vpc-d3a77cb6

Cancel Yes, Create

- d. Lorsque les paramètres vous conviennent, choisissez Yes, Create.
8. Définissez une règle de trafic entrant réseau pour votre groupe de sécurité. Cette règle permet de vous connecter à votre instance Amazon EC2 à l'aide du protocole SSH (Secure Shell).
- a. Dans 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 \(console\)](#).

Connexion à un cache en cours d'exécution dans un réseau Amazon VPC

Cet exemple indique comment lancer une instance Amazon EC2 dans votre Amazon VPC. Vous pouvez ensuite vous connecter à cette instance et accéder au cache ElastiCache qui s'exécute dans le réseau Amazon VPC.

Connexion à un cache en cours d'exécution dans un réseau Amazon VPC (console)

Dans cet exemple, vous allez créer une instance Amazon EC2 dans votre Amazon VPC. Vous pouvez utiliser cette instance Amazon EC2 pour vous connecter aux nœuds de cache s'exécutant dans l'Amazon VPC.

Note

Pour plus d'informations sur l'utilisation d'Amazon EC2, veuillez consulter le [Guide de mise en route d'Amazon EC2](#) dans la [Documentation Amazon EC2](#).

Pour créer une instance Amazon EC2 dans votre Amazon VPC à l'aide de la console Amazon EC2

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon EC2 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.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances

Purchasing option Request Spot Instances

Network

Subnet
250 IP Addresses available

Public IP Automatically assign a public IP address to your instances

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 Etiqueter l'instance, saisissez un nom pour votre instance Amazon EC2, 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 de plus amples informations sur les groupes de sécurité, veuillez consulter [Groupes de sécurité Amazon EC2 pour les instances Linux](#).

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-1a3d2178	default	default VPC security group
<input checked="" type="checkbox"/> sg-f13d2193	my-vpc-security-group	Testing

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, veuillez consulter le [Guide de mise en route d'Amazon EC2](#).

11. Lorsque vous êtes prêt à lancer votre instance Amazon EC2, choisissez Launch (Lancer).

Vous pouvez maintenant attribuer une adresse IP à l'instance Amazon EC2 que vous venez de créer. Vous devez utiliser cette adresse IP pour vous connecter à l'instance Amazon EC2.

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, choisissez IP Elastic.
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 Associate Address (Associer l'adresse), dans la zone Instance, sélectionnez l'ID de l'instance Amazon EC2 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'instance Amazon EC2 à l'aide de l'adresse IP élastique que vous avez créée.

Pour vous connecter à votre instance Amazon EC2

- 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 instance Amazon EC2.

Vous êtes maintenant prêt à interagir avec votre cluster ElastiCache. 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)

1. 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!
```

2. Accédez à la console ElastiCache à l'adresse <https://console.aws.amazon.com/elasticache/> afin d'obtenir le point de terminaison de l'un des nœuds dans votre cluster de cache. Pour plus d'informations, voir [Recherche de points de terminaison de connexion](#) pour Memcached.
3. 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
```

Important

Pour éviter de payer des frais supplémentaires sur votre compte AWS, veuillez à supprimer toutes les ressources AWS que vous ne souhaitez plus après avoir essayé ces exemples.

API Amazon ElastiCache et points de terminaison d'un VPC d'interface (AWS PrivateLink)

Vous pouvez établir une connexion privée entre votre VPC et vos points de terminaison d'API Amazon ElastiCache en créant un point de terminaison d'un VPC d'interface. Les points de terminaison d'interface sont optimisés par la technologie [AWS PrivateLink](#). AWS PrivateLink vous permet d'accéder en privé aux opérations d'API Amazon ElastiCache sans passerelle Internet, dispositif NAT, connexion VPN ou connexion AWS Direct Connect.

Les instances de votre VPC ne nécessitent pas d'adresse IP publique pour communiquer avec les points de terminaison d'API Amazon ElastiCache. Vos instances ne nécessitent pas non plus d'adresse IP publique pour utiliser l'une des opérations d'API ElastiCache 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 Elastic, veuillez consulter [Interfaces réseau Elastic](#) dans le Guide de l'utilisateur Amazon EC2.

- Pour plus d'informations sur les points de terminaison d'un VPC, veuillez consulter [Points de terminaison d'un VPC d'interface \(AWS PrivateLink\)](#) dans le Guide de l'utilisateur Amazon VPC.
- Pour plus d'informations sur les opérations d'API ElastiCache, consultez [Opérations d'API ElastiCache](#).

Une fois que vous avez créé un point de terminaison d'un VPC d'interface, si vous activez les noms d'hôtes [DNS privés](#) pour le point de terminaison, le point de terminaison ElastiCache par défaut (<https://elasticache.Region.amazonaws.com>) est résolu vers votre point de terminaison d'un 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 de plus amples informations, consultez la rubrique [Points de terminaison d'un VPC d'interface \(AWS PrivateLink\)](#) du Guide de l'utilisateur Amazon VPC. ElastiCache prend en charge l'exécution d'appels en direction de toutes ses [actions d'API](#) à l'intérieur de votre VPC.

Note

Les noms d'hôtes DNS privés ne peuvent être activés que pour un seul point de terminaison d'un VPC dans le VPC. Si vous voulez créer un point de terminaison d'un VPC supplémentaire, le nom d'hôte DNS privé doit être désactivé pour celui-ci.

Considérations relatives aux points de terminaison d'un VPC

Avant de configurer un point de terminaison d'un VPC d'interface pour les points de terminaison d'API Amazon ElastiCache, assurez-vous de consulter [Propriétés et limitations des points de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC. Toutes les opérations de l'API ElastiCache pertinentes à la gestion des ressources Amazon ElastiCache sont disponibles à partir de votre VPC à l'aide d'AWS PrivateLink.

Les politiques de point de terminaison d'un VPC sont prises en charge pour les points de terminaison de l'API ElastiCache. Par défaut, l'accès complet aux opérations de l'API ElastiCache 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 d'un VPC d'interface pour l'API ElastiCache

Vous pouvez créer un point de terminaison d'un VPC pour l'API Amazon ElastiCache à l'aide de la console Amazon VPC ou d'AWS CLI. Pour de plus amples informations, veuillez consulter [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. Dans ce cas, le point de terminaison Amazon ElastiCache par défaut (<https://elasticache.Region.amazonaws.com>) est résolu vers votre point de terminaison d'un VPC. Pour les régions AWS Chine (Beijing) et Chine (Ningxia), vous pouvez effectuer des demandes d'API avec le point de terminaison d'un VPC en utilisant elasticache.cn-north-1.amazonaws.com.cn pour Beijing et elasticache.cn-northwest-1.amazonaws.com.cn 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 d'un VPC pour l'API Amazon ElastiCache

Vous pouvez attacher une politique de point de terminaison à votre point de terminaison d'un VPC qui contrôle l'accès à l'API ElastiCache. 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.

Exemple Politique de point de terminaison d'un VPC pour les actions de l'API ElastiCache

Voici un exemple de politique de point de terminaison pour l'API ElastiCache. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux actions de l'API ElastiCache énumérées pour tous les principaux sur toutes les ressources.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  }]
}
```

Exemple Politique de point de terminaison d'un VPC qui refuse tout accès à partir d'un compte AWS spécifié

La stratégie de point de terminaison VPC suivante refuse au compte AWS **123456789012** tout accès aux ressources utilisant le point de terminaison. 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 auto-conçu dans un réseau Amazon VPC, vous devez utiliser un groupe de sous-réseaux. ElastiCache utilise ce groupe de sous-réseaux pour choisir un sous-réseau et des adresses IP au sein de celui-ci à associer à vos nœuds.

ElastiCache fournit un groupe de sous-réseaux IPv4 par défaut ou vous pouvez choisir d'en créer un nouveau. Pour IPv6, vous devez créer un groupe de sous-réseaux avec un bloc d'adresse CIDR IPv6. Si vous choisissez dual stack (double pile), vous devez sélectionner un Discovery IP type (Type d'adresse IP de découverte), IPv6 ou IPv4.

ElastiCache sans serveur n'utilise aucune ressource de groupe de sous-réseaux, mais utilise directement une liste de sous-réseaux lors de la création.

Cette section décrit comment créer et tirer parti des sous-réseaux et des groupes de sous-réseaux pour gérer l'accès à vos ressources ElastiCache.

Pour plus d'informations sur l'utilisation du groupe de sous-réseaux dans un environnement Amazon VPC, veuillez consulter [Accès à votre cluster](#).

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. Ensuite, ElastiCache utilise ce groupe de sous-réseaux de cache pour attribuer des adresses IP à chaque nœud de cache dans le cache, au sein de ce sous-réseau.

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 comme 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 des adresses IP au sein de celui-ci à associer à vos nœuds. Si vous choisissez la double pile ou IPv6, vous serez redirigé vers la création de sous-réseaux à double pile ou IPv6. Pour plus d'informations sur les types de réseaux, consultez [Type de réseau](#). 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), la AWS CLI et l'API ElastiCache.

Pour créer un groupe de sous-réseaux (console)

La procédure suivante indique comment créer un groupe de sous-réseaux (console).

Pour créer un groupe de sous-réseaux (console)

1. Connectez-vous à la Console de gestion AWS et ouvrez la console ElastiCache à l'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](#) et les ID de 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 Subnet Groups (Groupes de sous-réseaux) de la console ElastiCache. 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 de plus amples informations, consultez la rubrique AWS CLI [create-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.

Modification d'un groupe de sous-réseaux

Vous pouvez modifier la description d'un groupe de sous-réseaux de , ou modifier la liste des ID de sous-réseaux associés au groupe de sous-réseaux de 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 AWS Management Console et ouvrez la console ElastiCache 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 de plus amples informations, consultez la rubrique AWS CLI [modify-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 AWS Management Console et ouvrez la console ElastiCache 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 de plus amples informations, consultez la rubrique AWS CLI [delete-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 stratégies basées sur l'identité pour Amazon ElastiCache](#)
- [Résolution des problèmes d' ElastiCache identité et d'accès à Amazon](#)
- [Contrôle d'accès](#)
- [Présentation de la gestion des autorisations d'accès à vos ressources ElastiCache](#)

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, voir [Comment Amazon ElastiCache travaille avec IAM](#).

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaitez peut-être en savoir plus sur la manière dont vous pouvez rédiger 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 stratégies basées sur l'identité pour Amazon ElastiCache](#).

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS à 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 à l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section [Comment vous connecter à votre compte](#) [Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d'AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section [Signature des demandes AWS d'API](#) dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir

plus, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez [Tâches nécessitant des informations d'identification d'utilisateur root](#) dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède en utilisant les 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 votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous

vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez [Quand créer un utilisateur IAM \(au lieu d'un rôle\)](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux [ressources entre comptes dans IAM dans le guide de l'utilisateur IAM](#).
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
 - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).
 - Rôle de service : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
 - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un Service AWS. Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage

des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces

politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez [Vue d'ensemble des listes de contrôle d'accès \(ACL\)](#) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- **Politiques de contrôle des services (SCP)** — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée des comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations .
- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez [politiques de séance](#) dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser

une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

Comment 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

Fonction 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 d'une politique	Oui
ACL	Oui
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Fonctions de service	Oui
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble 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 [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de stratégies basées sur l'identité pour ElastiCache

Pour consulter des exemples de politiques ElastiCache basées sur l'identité, consultez. [Exemples de stratégies 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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les

ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez [la section Accès aux ressources entre comptes dans IAM](#) dans le guide de l'utilisateur d'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 stratégie 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 stratégies 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 ARN, 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 stratégies 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 stratégies basées sur l'identité pour Amazon ElastiCache](#)

Listes de contrôle d'accès (ACL) dans ElastiCache

Prend en charge les listes ACL	Oui
--------------------------------	-----

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Contrôle d'accès basé sur les attributs (ABAC) avec ElastiCache

Prend en charge ABAC (étiquettes dans les politiques)	Oui
---	-----

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans [l'élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez [Qu'est-ce que le contrôle d'accès basé sur les attributs \(ABAC\) ?](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec 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 [Changement de rôle \(console\)](#) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez [Informations d'identification de sécurité temporaires dans IAM](#).

Autorisations de principal 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 fonctions du service Oui

Une fonction de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un 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. Modifiez les rôles de service uniquement lorsque ElastiCache vous recevez des instructions à cet effet.

Rôles liés à un service pour ElastiCache

Prend en charge les rôles liés à un service. 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 au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#). Recherchez un service dans le tableau qui inclut un Yes dans la

colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de stratégies basées sur l'identité pour Amazon ElastiCache

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources ElastiCache. Ils ne peuvent pas non plus exécuter des tâches à l'aide de la AWS Management Console, de l'AWS Command Line Interface (AWS CLI) ou de l'API AWS. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM doit 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](#) 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 des ARN pour chaque type de ressource, consultez [Actions, ressources et clés de condition pour Amazon ElastiCache](#) dans la Référence de l'autorisation de service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console ElastiCache](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

Bonnes pratiques en matière de politiques

Les stratégies basées sur l'identité déterminent si une personne peut créer, consulter ou supprimer des ressources ElastiCache 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 :

- Démarrer avec AWS gérées et évoluez vers les autorisations de moindre privilège - Pour commencer à accorder des autorisations à vos utilisateurs et charges de travail, utilisez les politiques gérées AWS qui accordent des autorisations dans de nombreux cas d'utilisation courants. Elles sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire encore les autorisations en définissant des Politiques gérées par le client AWS qui sont spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [Politiques gérées AWS](#) ou [Politiques gérées AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.

- Accorder les autorisations de moindre privilège - Lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez [Politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès - Vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées via un Service AWS spécifique, comme AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles - IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politique IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Authentification multifactorielle (MFA) nécessaire : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root dans votre Compte AWS, activez l'authentification multifactorielle pour une sécurité renforcée. Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

Utilisation de la console ElastiCache

Pour accéder à la console Amazon ElastiCache, vous devez disposer d'un jeu minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources ElastiCache de votre Compte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Vous n'avez pas besoin d'accorder les autorisations minimales de console pour les utilisateurs qui effectuent des appels uniquement à AWS CLI ou à l'API AWS. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour vous assurer que les utilisateurs et les rôles puissent continuer à utiliser la console ElastiCache, attachez également la politique ElastiCache ConsoleAccess ou ReadOnly gérée par AWS 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 nécessaires pour réaliser cette action sur la console ou par programmation à l'aide de l'AWS CLI ou de l'API AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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 un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole` action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle ElastiCache.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans ElastiCache. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes 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 (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises ElastiCache en charge, consultez [Comment 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 connaître la différence entre l'utilisation de rôles et de politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux [ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM](#).

Contrôle d'accès

Vous pouvez disposer d'informations d'identification valides pour authentifier vos demandes, mais vous ne pouvez pas créer de 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.

- [Présentation de la gestion des autorisations d'accès à vos ressources ElastiCache](#)
- [Utilisation des politiques basées sur l'identité \(politiques IAM\) pour Amazon ElastiCache](#)

Présentation de la gestion des autorisations d'accès à vos ressources ElastiCache

Chaque ressource AWS appartient à un compte AWS et les autorisations de créer des ressources et d'y accéder 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 ElastiCache prend également en charge l'attachement de politiques d'autorisations aux ressources.

Note

Un administrateur de compte (ou utilisateur administrateur) est un utilisateur doté des privilèges d'administrateur. Pour plus d'informations, consultez [Bonnes pratiques IAM](#) dans le Guide de l'utilisateur IAM.

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

- Utilisateurs et groupes dans AWS IAM Identity Center :

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

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

Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) du Guide de l'utilisateur IAM.

- Utilisateurs IAM :

- Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique [Création d'un rôle pour un utilisateur IAM](#) du Guide de l'utilisateur IAM.

- (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique [Ajout d'autorisations à un utilisateur \(console\)](#) du Guide de l'utilisateur IAM.

Rubriques

- [Ressources et opérations Amazon ElastiCache](#)
- [Présentation de la propriété des ressources](#)

- [Gestion de l'accès aux ressources](#)
- [Politiques gérées par AWS pour Amazon ElastiCache](#)
- [Utilisation des politiques basées sur l'identité \(politiques IAM\) pour Amazon ElastiCache](#)
- [Autorisations de niveau ressource](#)
- [Utilisation de clés de condition](#)
- [Utilisation des rôles liés à un service pour Amazon ElastiCache](#)
- [ElastiCache Autorisations d'API : référence aux actions, aux ressources et aux conditions](#)

Ressources et opérations Amazon ElastiCache

Pour afficher la liste des types de ressources ElastiCache et de leurs ARN, consultez [Ressources définies par Amazon ElastiCache](#) dans la Référence de l'autorisation de service. Pour connaître les actions avec lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon ElastiCache](#).

Présentation de la propriété des ressources

Le propriétaire de la ressource est le compte AWS qui l'a créée. Ainsi, le propriétaire de la ressource est le compte AWS de l'entité principale qui authentifie la demande ayant créé 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 utilisez les informations d'identification de compte racine de votre compte AWS pour créer un cluster de cache. Dans ce cas, votre compte AWS est le propriétaire de la ressource. Dans ElastiCache, la ressource est le cluster de cache.
- Supposons que vous créez un utilisateur IAM dans votre compte AWS et que vous accordez à cet utilisateur les autorisations permettant de créer un cluster de cache. Dans ce cas, l'utilisateur peut créer un cluster de cache. Toutefois, votre compte AWS, auquel l'utilisateur appartient, détient la ressource du cluster de cache.
- Supposons que vous créez un rôle IAM dans votre compte AWS avec les autorisations pour créer un cluster de cache. Dans ce cas, toute personne pouvant endosser le rôle peut créer un cluster de cache. Votre compte AWS, auquel le rôle appartient, détient 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'autorisation.

Note

Cette section traite de 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 ressource ElastiCache, 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 afin

d'accorder des autorisations inter-comptes à un autre compte AWS (par exemple, le Compte B) ou à un service AWS 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 alors déléguer des autorisations pour assumer ce rôle auprès de tous les utilisateurs figurant dans le compte B. Les utilisateurs du compte B sont ainsi autorisés à créer des ressources ou à y accéder dans le compte A. Dans certains cas, vous pourrez vouloir accorder à un service AWS des autorisations pour assumer ce 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 autorisant un utilisateur à effectuer l'action `DescribeCacheClusters` pour votre compte AWS. ElastiCache prend également en charge l'identification de ressources spécifiques à l'aide des ARN de ressource pour les actions d'API. (Cette approche est également appelée autorisations au niveau des ressources.)

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DescribeCacheClusters",
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeCacheClusters"
    ],
    "Resource": resource-arn
  }
]
```

Pour plus d'informations sur l'utilisation des politiques basées sur l'identité avec ElastiCache, veuillez consulter [Utilisation des 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 ressource Amazon ElastiCache (voir [Ressources et opérations Amazon ElastiCache](#)), le service définit un ensemble d'opérations d'API (voir [Actions](#)). Pour accorder des autorisations pour ces opérations d'API, ElastiCache définit un ensemble d'actions que vous pouvez spécifier dans une politique. Par exemple, pour la ressource de cluster ElastiCache, les actions suivantes sont définies : `CreateCacheCluster`, `DeleteCacheCluster` 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 [Ressources et opérations Amazon ElastiCache](#).
- **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 l'Effect spécifié, l'autorisation `elasticache:CreateCacheCluster` accorde ou refuse à l'utilisateur les autorisations pour effectuer l'opération `CreateCacheCluster` Amazon ElastiCache.
- **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 accéder à un tableau répertoriant toutes les actions d'API Amazon ElastiCache, veuillez consulter [ElastiCache 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 les clés de condition spécifiques à ElastiCache, veuillez consulter [Utilisation de clés de condition](#). Il existe des clés de condition à l'échelle d'AWS que vous pouvez utiliser selon vos besoins. Pour obtenir la liste complète des clés à l'échelle d'AWS, consultez [Clés disponibles pour les conditions](#) dans le Guide de l'utilisateur IAM.

Politiques gérées par AWS pour Amazon ElastiCache

Une politique gérée par AWS est une politique autonome créée et administrée par AWS. Les politiques gérées par AWS 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.

Gardez à l'esprit que les politiques gérées par AWS peuvent ne pas accorder les autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont disponibles pour tous les clients AWS. Nous vous recommandons de réduire encore les autorisations en définissant des [politiques gérées par le client](#) qui sont spécifiques à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques gérées par AWS. Si AWS met à jour les autorisations définies dans une politique gérée par AWS, la mise à jour affecte toutes les identités de principal (utilisateurs, groupes et rôles) auxquelles la politique est associée. AWS est plus susceptible de mettre à jour une politique gérée par AWS lorsqu'un nouveau Service AWS est lancé ou que de nouvelles opérations API deviennent accessibles pour les services existants.

Pour plus d'informations, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

Politique gérée par AWS : ElastiCacheServiceRolePolicy

Vous ne pouvez pas attacher ElastiCacheServiceRolePolicy à vos entités IAM. Cette politique est attachée à un rôle lié au service qui permet à ElastiCache d'effectuer des actions en votre nom.

Cette politique permet à ElastiCache de gérer des ressources AWS en votre nom selon les besoins, en vue de gérer votre cache :

- `ec2` : gère les ressources réseau EC2 à attacher aux nœuds de cache, notamment les points de terminaison de VPC (pour les caches sans serveur), les interfaces réseau Elastic (ENI) (pour les clusters auto-conçus) et les groupes de sécurité.
- `cloudwatch` : émet des données de métriques à partir du service dans CloudWatch.
- `outposts` : autorise la création de nœuds de cache sur AWS Outposts.

La politique [ElastiCacheServiceRolePolicy](#) est disponible sur la console IAM et vous pouvez consulter [ElastiCacheServiceRolePolicy](#) dans le Guide de référence des politiques gérées par AWS.

Politique gérée par AWS : AmazonElastiCacheFullAccess

Vous pouvez associer la politique AmazonElastiCacheFullAccess à vos identités IAM.

Cette politique fournit aux principaux un accès complet à ElastiCache à l'aide de la console de gestion AWS :

- `elasticache` : accède à toutes les API.
- `iam` : crée le rôle lié au service nécessaire au fonctionnement du service.
- `ec2` : décrit les ressources EC2 dépendantes nécessaires à la création du cache (VPC, sous-réseau, groupe de sécurité) et autorise la création de points de terminaison de VPC (pour les caches sans serveur).
- `kms` : autorise l'utilisation de clés CMK gérées par le client pour le chiffrement au repos.
- `cloudwatch` : autorise l'accès aux métriques pour afficher les métriques ElastiCache 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` : permet de remplir AWS Outposts pour la création de caches dans la console.
- `sns` : permet de remplir les rubriques SNS pour la fonctionnalité de notification dans la console.

La politique [AmazonElastiCacheFullAccess](#) est disponible dans la console IAM et vous pouvez consulter [AmazonElastiCacheFullAccess](#) dans le Guide de référence des politiques gérées par AWS.

Politique gérée par AWS : AmazonElastiCacheReadOnlyAccess

Vous pouvez associer la politique AmazonElastiCacheReadOnlyAccess à vos identités IAM.

Cette politique fournit aux principaux un accès en lecture seule à ElastiCache à l'aide de la console de gestion AWS :

- `elasticache` : accède aux API `Describe` en lecture seule.

La politique [AmazonElastiCacheReadOnlyAccess](#) est disponible dans la console IAM et vous pouvez consulter [AmazonElastiCacheReadOnlyAccess](#) dans le Guide de référence des politiques gérées par AWS.

Mises à jour des politiques gérées par AWS pour ElastiCache

Consultez les détails des mises à jour apportées aux politiques gérées par AWS pour 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 document ElastiCache.

Modification	Description	Date
AmazonElastiCacheFullAccess : mise à jour d'une politique existante	ElastiCache a ajouté de nouvelles autorisations pour permettre la gestion des caches sans serveur et 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 de VPC pour les ressources de cache sans serveur.	27 novembre 2023
ElastiCache a commencé à suivre les modifications	ElastiCache a commencé à suivre les modifications pour ses politiques gérées par AWS.	7 février 2020

Utilisation des 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

En premier lieu, nous vous recommandons de lire les rubriques qui détaillent les concepts de base et les options disponibles pour gérer l'accès à vos ressources Amazon ElastiCache. Pour de plus amples informations, veuillez consulter [Présentation de la gestion des autorisations d'accès à vos ressources ElastiCache](#).

Les sections de cette rubrique couvrent les sujets suivants :

- [Politiques gérées par AWS pour Amazon ElastiCache](#)
- [Exemples de politiques gérées par le client](#)

Un exemple de politique d'autorisation est exposé ci-dessous.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowClusterPermissions",
    "Effect": "Allow",
    "Action": [
      "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 instruction accorde des autorisations pour les actions Amazon ElastiCache (`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 un tableau présentant toutes les actions de l'API Amazon ElastiCache et les ressources auxquelles elles s'appliquent, veuillez consulter [ElastiCache 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 : Permettre à un utilisateur d'appeler l'API IAM CreateServiceLinkedRole](#)). Ou vous devez avoir créé un rôle lié à un service ElastiCache.

Lorsqu'ils sont combinés avec les autorisations minimales nécessaires à l'utilisation de la console Amazon ElastiCache, les exemples de politiques dans cette section accordent des autorisations supplémentaires. Les exemples s'appliquent également à la AWS CLI et aux kits SDK AWS.

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. Certains actions ElastiCache qui peuvent paraître simples peuvent nécessiter d'autres actions pour les prendre en charge lorsque vous utilisez la console ElastiCache. Par exemple, `elasticache:CreateCacheCluster` donne des autorisations pour créer des clusters de cache ElastiCache. Toutefois, afin d'effectuer cette opération, la console ElastiCache utilise un certain nombre d'actions `List` et `Describe` afin de remplir les listes de la console.

Exemples

- [Exemple 1 : Accorder à un utilisateur un accès en lecture seule aux ressources ElastiCache](#)
- [Exemple 2 : Permettre à un utilisateur d'effectuer les tâches courantes d'administrateur système ElastiCache](#)
- [Exemple 3 : Permettre à un utilisateur d'accéder à toutes les actions d'API ElastiCache](#)
- [Exemple 4 : Permettre à un utilisateur d'appeler l'API IAM `CreateServiceLinkedRole`](#)

Exemple 1 : Accorder à un utilisateur un accès en lecture seule aux ressources ElastiCache

La politique suivante accorde à un utilisateur l'autorisation d'utiliser les actions ElastiCache pour afficher les ressources. En général, vous attachez ce type de politique d'autorisations à un groupe de gestionnaires.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECReadOnly",
    "Effect": "Allow",
    "Action": [
      "elasticache:Describe*",
      "elasticache:List*"
    ],
    "Resource": "*"
  }
]
```

Exemple 2 : Permettre à un utilisateur d'effectuer les tâches courantes d'administrateur système ElastiCache

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 actions ElastiCache pour ces tâches d'administrateur système courantes. Généralement, vous attachez ce type de politique d'autorisations au groupe d'administrateurs système.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowMutations",
    "Effect": "Allow",
    "Action": [
      "elasticache:Modify*",
      "elasticache:Describe*",
      "elasticache:ResetCacheParameterGroup"
    ],
    "Resource": "*"
  }
]
```

Exemple 3 : Permettre à un utilisateur d'accéder à toutes les actions d'API ElastiCache

La politique suivante permet à un utilisateur d'accéder à toutes les actions ElastiCache. Nous vous conseillons d'accorder ce type de politique d'autorisations uniquement à un utilisateur administrateur.

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

Exemple 4 : Permettre à un utilisateur d'appeler l'API IAM CreateServiceLinkedRole

La politique suivante permet à un utilisateur d'appeler l'API CreateServiceLinkedRole IAM. Nous vous conseillons d'accorder ce type de politique d'autorisation à l'utilisateur qui appelle les opérations de modification ElastiCache.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "elasticache.amazonaws.com"
        }
      }
    }
  ]
}
```

Autorisations de niveau ressource

Vous pouvez restreindre la portée des autorisations en spécifiant des ressources dans une politique IAM. De nombreuses actions d'API ElastiCache 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 afficher la liste des types de ressources ElastiCache et de leurs ARN, consultez [Ressources définies par Amazon ElastiCache](#) dans la Référence de l'autorisation de service. Pour connaître les actions avec lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon ElastiCache](#).

Exemples

- [Exemple 1 : accorder à un utilisateur un accès complet aux ressources ElastiCache spécifiques](#)
- [Exemple 2 : Refuser à un utilisateur l'accès à un cache sans serveur.](#)

Exemple 1 : accorder à un utilisateur un accès complet aux ressources ElastiCache 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'élément `Condition` d'une politique JSON pour comparer des clés dans le contexte de demande avec les valeurs de clé spécifiées dans votre politique. Pour plus d'informations, consultez [Éléments de politique JSON IAM : condition](#).

Pour obtenir la liste des clés de condition ElastiCache, consultez [Clés de condition pour Amazon ElastiCache](#) dans la Référence de l'autorisation de service.

Pour obtenir la liste de toutes les clés de condition globales, veuillez consulter [Clés de contexte de condition globales AWS](#).

Spécification de conditions : Utilisation de clés de condition

Pour mettre en 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`.

```
{
  "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 [Tag-Based access control policy examples](#).

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 pour la mise en œuvre d'un contrôle d'accès affiné sur les paramètres ElastiCache répertoriés 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.

```
{
  "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:MaximumECPUPerSecond` : 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 pouvant exécuter un nombre d'ECPU par seconde supérieur au nombre spécifié.

```

{
  "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 la ou les valeurs `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.

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

4. `elasticache:EngineVersion` : spécifiez l'utilisation de la version du moteur 1.6.6.

```
{
```

```

"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"
      }
    }
  }
]
}

```

5. `elasticache:KmsKeyId` : spécifiez l'utilisation de clés KMS AWS gérées par le client.

```

{
  "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"
        }
    }
}
]
}

```

6. `elasticache:CacheParameterGroupName` : spécifiez un groupe de paramètres non 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 d'utilisation limitée de seulement « my-org-param-group ».

```

{
    "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"
      }
    }
  }
]
}

```

7. `elasticache:CreateCacheCluster` : Refus de l'action `CreateCacheCluster` si l'identification de la demande `Project` est manquante ou n'est pas égale à `Dev`, `QA` ou `Prod`.

```

{
  "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"
            ]
        }
    }
}
]
}

```

8. `elasticache:CacheNodeType` : autorise `CreateCacheCluster` avec `cacheNodeType` `cache.r5.large` ou `cache.r6g.4xlarge` et la balise `Project=XYZ`.

```

{
    "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 des rôles liés à un service pour Amazon ElastiCache

Amazon ElastiCache utilise les [rôles liés à un service](#) AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à un service AWS, comme Amazon ElastiCache. Les rôles liés au service Amazon ElastiCache 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 simplifie la configuration d'Amazon ElastiCache, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. Ces rôles se trouvent dans votre compte AWS, mais sont liés à des cas d'utilisation Amazon ElastiCache et disposent d'autorisations prédéfinies. Seul Amazon ElastiCache peut endosser ces rôles et seuls ces rôles peuvent utiliser la politique

d'autorisations prédéfinie. Vous pouvez supprimer les rôles uniquement après la suppression préalable de leurs ressources connexes. Vos ressources Amazon ElastiCache sont ainsi protégées, car vous ne pouvez pas involontairement supprimer les autorisations nécessaires pour accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez [AWS Services qui fonctionnent avec IAM](#) et recherchez les services avec un Oui dans la colonne rôle lié au service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Table des matières

- [Autorisations du rôle lié 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 \(API IAM\)](#)
- [Suppression d'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 du rôle lié à un service pour Amazon ElastiCache

Autorisations pour créer un rôle lié à un service

Pour permettre à une entité IAM de créer un rôle lié à un service AWSServiceRoleForElastiCache

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/
AWSServiceRoleForElastiCache*",
"Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

Pour permettre à une entité IAM de supprimer un rôle lié à un service `AWSServiceRoleForElastiCache`

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/
AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

Vous pouvez également utiliser une politique gérée par AWS 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 AWS Management Console et ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
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 Or select a service to view its use cases (Ou sélectionner un service pour afficher ses cas d'utilisation, choisissez ElastiCache.
5. Sélectionnez Next: Permissions (Étape suivante : 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 de la AWS Command Line Interface pour supprimer 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 lié à un service `AWSServiceRoleForElastiCache`. 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 de la 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 courante d'un rôle, utilisez la AWS CLI pour l'opération IAM [get-role](#).

Exemple

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

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 la AWS CLI pour l'opération IAM [update-role-description](#).

Pour Linux, macOS ou Unix :

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForElastiCache \  
  --description "new description"
```

Pour Windows :

```
$ aws iam update-role-description ^  
  --role-name AWSServiceRoleForElastiCache ^  
  --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](#).

Exemple

```
https://iam.amazonaws.com/  
  ?Action=GetRole  
  &RoleName=AWSServiceRoleForElastiCache  
  &Version=2010-05-08  
  &AUTHPARAMS
```

2. Pour mettre à jour la description d'un rôle, utilisez l'opération d'API IAM [UpdateRoleDescription](#).

Exemple

```
https://iam.amazonaws.com/  
  ?Action=UpdateRoleDescription  
  &RoleName=AWSServiceRoleForElastiCache  
  &Version=2010-05-08  
  &Description="New description"
```

Suppression d'un rôle lié à un service pour Amazon ElastiCache

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée 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 automatiquement le rôle lié à un 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 pas de ressources (clusters) 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 AWS Management Console et ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation de la console IAM, sélectionnez Roles (Rôles). Choisissez ensuite le nom (et non pas la case à cocher correspondante) du rôle AWSServiceRoleForElastiCache.
3. Sur la page Récapitulatif du rôle sélectionné, choisissez l'onglet Access Advisor.
4. Dans l'onglet Access Advisor, consultez l'activité récente pour le rôle lié à un service.

Pour supprimer les ressources Amazon ElastiCache qui nécessitent AWSServiceRoleForElastiCache

- Pour supprimer un cluster, consultez les rubriques suivantes :
 - [Utilisation de AWS Management Console](#)
 - [Utilisation de AWS CLI](#)
 - [Utilisation de l'API ElastiCache](#)

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 AWS Management Console et ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
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, vérifiez les dernières données consultées dans le service. Elles indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service AWS. Cela vous permet de confirmer si le rôle est actif actuellement. 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 de la AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (CLI)

1. Si vous ne connaissez pas le nom du rôle lié à un service que vous souhaitez supprimer, saisissez la commande suivante. Cette commande répertorie les rôles et leurs noms Amazon Resource Name (ARN) dans votre compte.

```
$ aws iam get-role --role-name role-name
```

Utilisez le nom du rôle, pas l'ARN, pour faire référence aux opérations de la CLI. Par exemple, si un rôle a l'ARN `arn:aws:iam::123456789012:role/myrole`, vous faites référence au rôle en tant que **myrole**.

2. Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Saisissez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Tapez la commande suivante pour vérifier l'état de la tâche de suppression.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

L'état de la tâche de suppression peut être NOT_STARTED, IN_PROGRESS, SUCCEEDED ou FAILED. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Suppression d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (API)

1. Pour envoyer une demande de suppression pour un rôle lié à un service, appelez [DeleteServiceLinkedRole](#). Dans la demande, spécifiez le nom d'un rôle.

Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le DeletionTaskId de la réponse afin de vérifier l'état de la tâche de suppression.

2. Pour vérifier l'état de la suppression, appelez [GetServiceLinkedRoleDeletionStatus](#). Dans la demande, spécifiez le DeletionTaskId.

L'état de la tâche de suppression peut être NOT_STARTED, IN_PROGRESS, SUCCEEDED ou FAILED. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

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, consultez [Utilisation 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 :

- [Guides de démarrage rapide sur la sécurité et la conformité](#) : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- [Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

 Note

Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le [HIPAA Eligible Services Reference](#).

- 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.

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 PCI DSS](#)

Résilience dans Amazon ElastiCache

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

Pour plus d'informations sur les régions et les zones de disponibilité AWS, consultez [AWS Infrastructure mondiale](#).

Outre l'infrastructure mondiale AWS, Amazon ElastiCache propose plusieurs fonctionnalités qui contribuent à la compatibilité avec vos besoins en matière de résilience et de sauvegarde de 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](#)
- [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.

[Si vous avez besoin de plus de 60 nœuds dans un cluster Memcached, ou de plus de 300 nœuds au total dans une AWS région, remplissez le formulaire de demande d'augmentation de ElastiCache limite à l'adresse https://aws.amazon.com/contact-us/elasticache-node-limit-request/.](https://aws.amazon.com/contact-us/elasticache-node-limit-request/)

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 l'éventualité peu probable d'un tel échec, vous perdrez les données mises en cache dans cette zone, et pas les données mises en cache dans les autres zones de disponibilité.

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 [Régions et zones de disponibilité](#).

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

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. La répartition de vos nœuds de manière uniforme dans les zones de disponibilité réduira l'impact d'une défaillance peu probable des zones de disponibilité. Ce processus est automatique pour les caches sans serveur.

Sécurité de l'infrastructure dans AWS ElastiCache

En tant que service géré, AWS ElastiCache est protégé par les procédures de sécurité réseau globales AWS décrites dans la section Sécurité et conformité du [Centre d'architecture AWS](#).

Vous utilisez les appels d'API publiés par AWS pour accéder à 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.

Gestion des mises à jour des services

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 :

Rubriques

- [Application des mises à jour de service](#)
- [Vérification de l'application de la dernière mise à jour de service à l'aide de la AWS console](#)
- [Gestion des mises à jour du service](#)

Application des mises à jour de service

Vous pouvez commencer à appliquer les mises à jour de service à votre flotte à partir du moment où les mises à jour ont un statut available (disponible). Les mises à jour de service sont cumulatives. En d'autres termes, toutes les mises à jour que vous n'avez pas encore appliquées sont incluses dans votre dernière mise à jour.

Si la mise à jour automatique est activée pour une mise à jour de service, vous pouvez choisir de ne rien faire 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.
 - **Date limite d'application recommandée** : ElastiCache date limite pour appliquer les mises à jour avant.
 - **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 plus d'informations, consultez [describe-service-updates](#).

- Pour appliquer une mise à jour de service à une liste de clusters, exécutez la commande suivante :

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

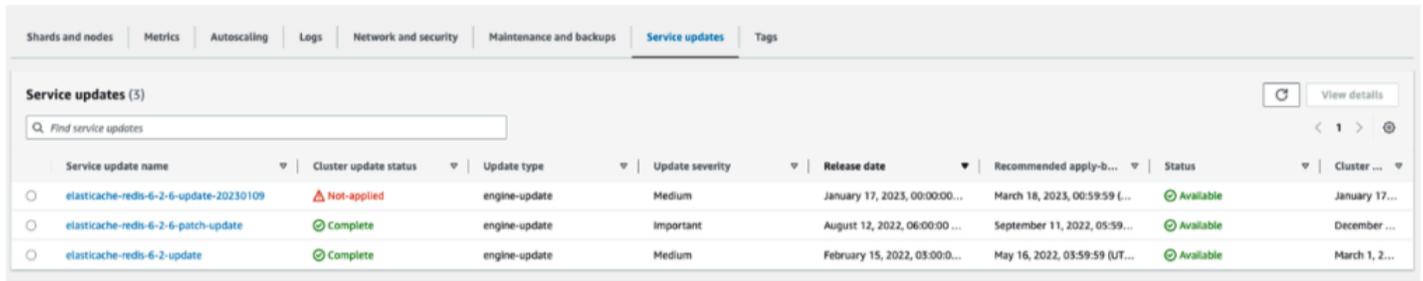
Pour plus d'informations, consultez [batch-apply-update-action](#).

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 exécutent la dernière mise à jour du service en procédant comme suit :

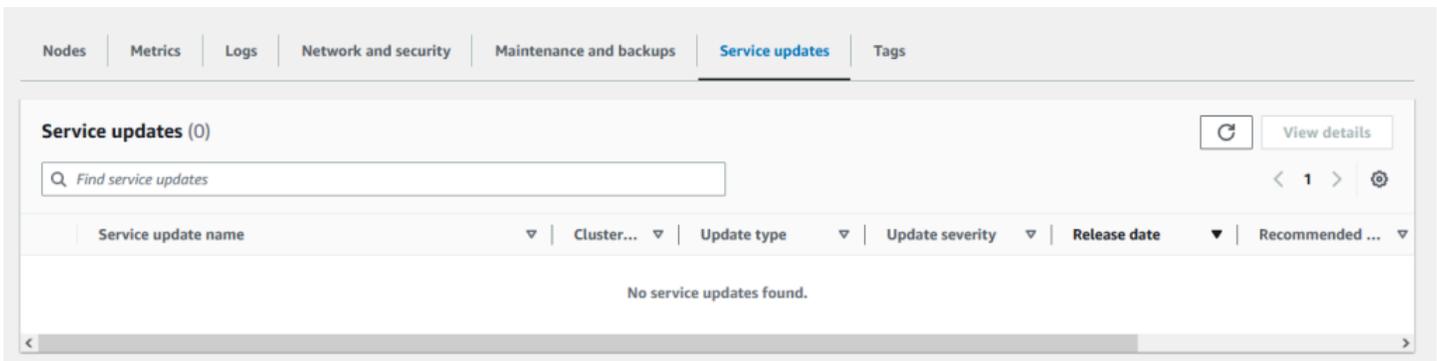
1. Choisissez un cluster applicable sur la page Redis Clusters
2. Choisissez Mises à jour de service dans le volet de navigation pour voir les mises à jour de service applicables à ce cluster, le cas échéant.

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.



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-by	Status	Cluster ...
elasticache-redis-6-2-6-update-202310109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 05:59:59 (UT...	Available	March 1, 2...

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.



Service update name	Cluster...	Update type	Update severity	Release date	Recommended ...
No service updates found.					

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 plus d'informations, consultez [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 plus d'informations, consultez [BatchStopUpdateAction](#).

Journalisation et surveillance dans Amazon ElastiCache

Pour gérer votre cache, il est important que vous connaissiez son fonctionnement. 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 sont apportées aux ressources de votre cache (par exemple, un nouveau cache est créé ou un cache est supprimé).

Rubriques

- [Métriques et événements pour les caches sans serveur](#)
- [Métriques et événements des clusters auto-conçus](#)
- [Journalisation des appels d'API Amazon ElastiCache avec AWS CloudTrail](#)
- [Journalisation des appels d'API Amazon ElastiCache avec AWS CloudTrail](#)

Métriques et événements pour les caches sans serveur

Cette section décrit les métriques et les événements que vous pouvez surveiller lorsque vous utilisez des caches sans serveur.

Rubriques

- [Métriques pour les caches sans serveur](#)
- [Événements pour les caches sans serveur](#)

Métriques pour les caches sans serveur

L'espace de noms AWS/ElastiCache inclut les métriques CloudWatch suivantes pour vos caches sans serveur Memcached.

Métrique	Description	Unité
BytesUsedForCache	Nombre total d'octets utilisés par les données stockées dans votre cache.	Octets

Métrique	Description	Unité
ElastiCacheProcessingUnits	Nombre total d'ECPU (ElastiCacheProcessingUnits) consommées par les demandes 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
CurrConnections	Nombre de connexions client à votre cache.	Nombre
ThrottledCmds	Nombre de demandes qui ont été limitées par ElastiCache, car la mise à l'échelle de la charge de travail était plus rapide que celle d'ElastiCache.	Nombre
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
NetworkBytesIn	Nombre total d'octets transférés dans le cache.	Octets

Métrique	Description	Unité
NetworkBytesOut	Nombre total d'octets transférés 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 suivantes au niveau des commandes Memcached.

Métrique	Description	Unité
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	Nombre

Métrique	Description	Unité
	cache dans lesquelles la clé demandée a été trouvée.	
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 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
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

Métrique	Description	Unité
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
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 pour les caches sans serveur

ElastiCache consigne les événements relatifs 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 récupérer facilement les événements à partir du journal en utilisant la console ElastiCache, la commande AWS CLI `describe-events` ou l'action d'API ElastiCache `DescribeEvents`.

Vous pouvez choisir de surveiller, d'ingérer, de transformer et d'agir sur les événements ElastiCache à l'aide d'Amazon EventBridge. Pour en savoir plus, consultez le Guide d'utilisation d'Amazon EventBridge à l'adresse <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

Affichage des événements ElastiCache (console)

Pour afficher les événements à l'aide de la console ElastiCache :

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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.

Affichage des événements ElastiCache (AWS CLI)

Pour générer une liste des événements ElastiCache à l'aide d'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 pour les caches sans serveur

Detail-Type	Description	Unité	Source	Message
Cache créé	ARN du cache	création	serverless-cache	Le cache <nom-cache> est créé et prêt à être utilisé.

Detail-Type	Description	Unité	Source	Message
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de 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 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 création du cache <nom-cache>. Limite de quota atteinte pour la création d'un point de terminaison de VPC.

Detail-Type	Description	Unité	Source	Message
Échec de la création du cache	ARN du cache	échec	serverless-cache	Échec de création du cache <nom-cache>. Vous n'êtes pas autorisé à créer un point de terminaison de VPC.

Événements de mise à jour pour les caches sans serveur

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache mis à jour	ARN du cache	modification de configuration	serverless-cache	Groupes de sécurité mis à jour pour le cache <nom-cache>.
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. Échec de la mise à jour des groupes de sécurité.

Detail-Type	Liste de ressources	Catégorie	Source	Message
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. Échec de la mise à jour des groupes de sécurité 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 <nom-cache> a échoué. La mise à jour des groupes de sécurité a échoué, car les groupes de sécurité ne sont pas valides.

Événements de suppression pour les caches sans serveur

Detail-Type	Liste de ressources	Catégorie	Source	Message
Cache supprimé	ARN du cache	suppression	serverless-cache	Le cache <nom-cache> a été supprimé.

Événements relatifs aux limites d'utilisation pour les caches sans serveur

Detail-Type	Description	Unité	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 en raison d'une configuration non valide.

Métriques et événements des clusters auto-conçus

Cette section décrit les métriques, les événements et les journaux que vous pouvez vous attendre à voir lorsque vous utilisez des clusters auto-conçus.

Rubriques

- [Métriques pour les clusters auto-conçus](#)
- [Événements pour les clusters auto-conçus](#)
- [Surveillance de l'utilisation à l'aide de métriques CloudWatch](#)
- [Surveillance des événements ElastiCache avec Amazon SNS](#)

Métriques pour les clusters auto-conçus

Lorsque vous concevez vous-même des clusters, ElastiCache émet des métriques au niveau de chaque nœud, notamment des métriques au niveau de l'hôte et des métriques de cache.

Pour plus d'informations sur les métriques au niveau de l'hôte pour Memcached, consultez [Métriques au niveau de l'hôte](#).

Pour plus d'informations sur les métriques au niveau du nœud pour Memcached, consultez [Métriques pour Memcached](#).

Événements pour les clusters auto-conçus

ElastiCache consigne les événements relatifs à vos caches auto-conçus. Lorsque vous utilisez des clusters auto-conçus, vous pouvez consulter les événements de votre cluster dans la console ElastiCache, à l'aide d'AWS CLI ou à l'aide d'Amazon Simple Notification Service (SNS). Les événements d'un cluster auto-conçu 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 récupérer facilement les événements à partir du journal en utilisant la console ElastiCache, la commande AWS CLI `describe-events` ou l'action d'API ElastiCache `DescribeEvents`.

Affichage des événements ElastiCache (Console)

La procédure suivante affiche les événements à l'aide de la console ElastiCache.

Pour afficher les événements à l'aide de la console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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. À l'aide du Filtre, vous pouvez choisir d'afficher tous les événements ou uniquement ceux d'un type spécifique dans la liste des événements.

Affichage des événements ElastiCache (AWS CLI)

Pour générer une liste des événements ElastiCache à l'aide d'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 événements ElastiCache suivants déclenchent des notifications Amazon SNS. Pour de plus amples informations sur les détails de l'événement, veuillez consulter [Affichage des événements ElastiCache](#).

Nom de l'événement	Message	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	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 disponibles	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	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 : <i>cluster-name</i>	Un ou plusieurs paramètres du cluster de cache ont été modifiés.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioning	La mise en service d'un cluster de cache est terminée, et

Nom de l'événement	Message	Description
	Complete <i>cluster-name-0001-005</i>	les nœuds de cache dans le cluster de cache sont prêts à être utilisés.
ElastiCache:CacheClusterProvisioningFailed en raison d'un état de réseau incompatible	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Il y a eu une tentative de lancement d'un nouveau cluster de cache dans un Virtual Private Cloud (VPC) inexistant.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionnement du cluster de cache réalisé avec succès.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	Le dimensionnement du cluster de cache a échoué.

Nom de l'événement	Message	Description
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	L'un des événements suivants s'est produit : <ul style="list-style-type: none">• La liste des groupes de sécurité de cache autorisés à accéder au cluster de cache a été modifiée.• Un ou plusieurs nouveaux groupes de sécurité EC2 ont été autorisés à accéder aux groupes de sécurité de cache associés au cluster de cache.• Un ou plusieurs nouveaux groupes de sécurité EC2 ont été déçus de leur autorisation d'accès par les groupes de sécurité de cache associés au cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache a détecté que l'hôte exécutant un nœud de cache est endommagé ou inaccessible, et a commencé à remplacer le nœud de cache.</p> <div data-bbox="1068 495 1507 806"><p> Note</p><p>L'entrée DNS pour le nœud de cache remplacé n'est pas changée.</p></div> <p>Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Toutefois, certaines bibliothèques client de cache peuvent cesser d'utiliser le nœud de cache même après qu'ElastiCache ait remplacé le nœud de cache ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.</p>

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache a détecté que l'hôte exécutant un nœud de cache est endommagé ou inaccessible, et a terminé le remplacement du nœud de cache.</p> <div data-bbox="1068 541 1507 856"><p> Note</p><p>L'entrée DNS pour le nœud de cache remplacé n'est pas changée.</p></div> <p>Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Toutefois, certaines bibliothèques client de cache peuvent cesser d'utiliser le nœud de cache même après qu'ElastiCache ait remplacé le nœud de cache ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.</p>

Nom de l'événement	Message	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	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 (uniquement Redis)	ElastiCache:CertificateRenewalComplete	Le certificat Amazon d'une autorité de certification a été renouvelé avec succès.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Le groupe de réplication a bien été créé.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	La suppression d'un cluster de cache et de tous les nœuds de cache associés est terminée.
ElastiCache:FailoverComplete (uniquement Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	Basculement vers un nœud du réplica réussi.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Le nombre de réplicas dans le cluster a été augmenté.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Le processus d'ajout de réplicas à votre cluster a commencé.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Le remplacement d'un nœud de votre cluster qui était prévu a été annulé.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	Le remplacement d'un nœud de cluster a été reprogrammé dans le créneau indiqué dans la notification. Pour en savoir plus sur les mesures que vous pouvez prendre, consultez Remplacement des nœuds de cache pour Memcached .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Un nœud du cluster doit être remplacé pendant le créneau décrit dans la notification. Pour en savoir plus sur les mesures que vous pouvez prendre, consultez Remplacement des nœuds de cache pour Memcached .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nœud de cache a été supprimé du cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	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 (uniquement Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Un instantané de cache s'est terminé avec succès.
ElastiCache:SnapshotFailed (uniquement Redis)	SnapshotFailed : <i>cluster-name</i>	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.

Surveillance de l'utilisation à l'aide de métriques CloudWatch

ElastiCache propose des métriques qui vous permettent de contrôler vos clusters. Vous pouvez accéder à ces métriques via CloudWatch. Pour plus d'informations sur l'utilisation de CloudWatch, veuillez consulter la [documentation CloudWatch](#).

ElastiCache fournit deux métriques au niveau de l'hôte (par exemple, utilisation du CPU) et des métriques qui sont spécifiques au logiciel de moteur de cache (par exemple, réussites d'accès au

cache et échecs d'accès au cache). Ces métriques sont mesurées et publiées pour chaque nœud de cache toutes les 60 secondes.

Important

Vous devriez envisager de définir des alarmes CloudWatch sur certaines métriques clés, afin de recevoir une notification 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 Memcached](#)
- [Quelles métriques dois-je surveiller ?](#)
- [Surveillance des métriques de cluster et de nœud CloudWatch](#)

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.

Voir aussi

- [Métriques pour Memcached](#)

Métrique	Description	Unité
CPUUtilization	Le pourcentage d'utilisation de l'UC pour l'hôte entier.	Pourcentage
CPUCreditBalance	Nombre de crédits UC gagnés qu'une instance a accumulés depuis son lancement ou son démarrage. Pour les instances T2 Standard, le CPUCreditBalance inclut également le nombre de crédits de lancement qui ont été accumulés.	Crédits (minutes vCPU)

Métrique	Description	Unité
	<p>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'instance. 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.</p> <p>L'instance peut dépenser les crédits figurant dans le CPUCreditBalance pour dépasser le niveau de base de l'utilisation du CPU.</p> <p>Les métriques de crédits UC sont disponibles uniquement toutes les 5 minutes.</p>	
CPUCreditUsage	<p>Nombre de crédits UC dépensés par l'instance pour l'utilisation de l'UC. Par exemple, un crédit UC est équivalent à un processeur virtuel fonctionnant à 100 % d'utilisation pendant une minute ou une combinaison équivalente de processeurs virtuels, d'utilisation et de temps (par exemple, un processeur virtuel fonctionnant à 50 % d'utilisation pendant deux minutes, ou deux processeurs virtuels fonctionnant à 25 % d'utilisation pendant deux minutes).</p> <p>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.</p>	Crédits (minutes vCPU)

Métrique	Description	Unité
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
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
NetworkBandwidthIn AllowanceExceeded	Nombre de paquets formés parce que la bande passante agrégée entrante a dépassé le maximum de l'instance.	Nombre
NetworkConntrackAllowanceExceeded	Nombre de paquets formés parce que le suivi des connexions a dépassé le maximum de l'instance et que de nouvelles connexions n'ont pas pu être établies. Cela peut entraîner une perte de paquets pour le trafic vers ou en provenance de l'instance.	Nombre
NetworkBandwidthOut AllowanceExceeded	Nombre de paquets formés parce que la bande passante agrégée sortante a dépassé le maximum de l'instance.	Nombre

Métrique	Description	Unité
NetworkPacketsPerSecondAllowanceExceeded	Nombre de paquets formés parce que le PPS bidirectionnel a dépassé le maximum de l'instance.	Nombre
NetworkMaxBytesIn	Rafale maximale d'octets reçus par minute.	Octets
NetworkMaxBytesOut	Rafale maximale d'octets transmis par minute.	Octets
NetworkMaxPacketsIn	Rafale maximale de paquets reçus par minute.	Nombre
NetworkMaxPacketsOut	Rafale maximale de paquets transmis par minute.	Nombre
SwapUsage	Nombre de permutations utilisées sur l'hôte.	Octets

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	Unité
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

Métrique	Description	Unité
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
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

Métrique	Description	Unité
<code>CurrConnections</code>	<p>Nombre de connexions au cache à un instant donné. ElastiCache utilise deux à trois connexions pour surveiller le cluster.</p> <p>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 <code>Nodetype Specific Parameters</code> du groupe de paramètres applicable.</p> <p>Le nombre total de connexions est la somme des connexions client, des connexions à superviser et des connexions internes mentionnées ci-dessus.</p>	Nombre
<code>CurrItems</code>	Nombre d'éléments actuellement stockés dans le cache.	Nombre
<code>DecrHits</code>	Nombre de requêtes decrement reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
<code>DecrMisses</code>	Nombre de requêtes decrement reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre
<code>DeleteHits</code>	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée a été trouvée.	Nombre
<code>DeleteMisses</code>	Nombre de requêtes delete reçues par le cache dans lesquelles la clé demandée était introuvable.	Nombre

Métrique	Description	Unité
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	Unité
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

Métrique	Description	Unité
<code>CurrConfig</code>	Nombre actuel de configurations stockées.	Nombre
<code>EvictedUnfetched</code>	Nombre d'éléments valides expulsés du cache récemment utilisé qui n'ont jamais été touchés après leur définition.	Nombre
<code>ExpiredUnfetched</code>	Nombre d'éléments arrivés à expiration récupérés du cache récemment utilisé qui n'ont jamais été touchés après leur définition.	Nombre
<code>SlabsMoved</code>	Nombre total de pages de la section qui ont été déplacées.	Nombre
<code>TouchHits</code>	Nombre de clés qui ont été touchées et auxquelles une nouvelle date d'expiration a été affectée.	Nombre
<code>TouchMisses</code>	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	Unité
<code>NewConnections</code>	Nombre de nouvelles connexions reçues par le cache. Celui-ci est dérivé de la statistique Memcached <code>total_connections</code> en enregistrant la modification de <code>total_connections</code> sur une période de temps. Ce sera toujours au moins 1, en raison d'une connexion réservée à un ElastiCache.	Nombre
<code>NewItems</code>	Nombre de nouveaux éléments stockés par le cache. Celui-ci est dérivé de la statistique Memcached <code>total_items</code> en enregistrant la	Nombre

Métrique	Description	Unité
	modification de total_items sur une période de temps.	
UnusedMemory	<p>Quantité de mémoire non utilisée par les données. Celle-ci est dérivée des statistiques Memcached limit_maxbytes et bytes en soustrayant bytes de limit_maxbytes.</p> <p>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.</p> <p>Pour de plus amples informations, veuillez consulter Memcached item memory usage.</p>	Octets

Quelles métriques dois-je surveiller ?

Les métriques CloudWatch suivantes permettent de mieux connaître les performances ElastiCache. Dans la plupart des cas, nous vous recommandons de définir des alarmes CloudWatch pour ces métriques, afin de pouvoir prendre des actions correctives avant que des problèmes de performance ne se produisent.

Métriques pour la surveillance

- [CPUUtilization](#)
- [SwapUsage](#)
- [Evictions](#)
- [CurrConnections](#)

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](#).

Puisque Memcached est multi-thread, cette métrique peut atteindre jusqu'à 90 %. Si vous dépassez cette limite, augmentez votre cluster de cache à l'aide d'un type de nœud de cache plus important, ou ajoutez davantage de nœuds de cache.

SwapUsage

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 métrique CloudWatch `FreeableMemory` est proche de 0 (c. -à-d. inférieure à 100 Mo) ou si la métrique `SwapUsage` est supérieure à la métrique `FreeableMemory`, cela indique qu'un nœud est soumis à une pression de mémoire. Si cela se produit, nous vous recommandons d'augmenter la valeur du paramètre `ConnectionOverhead`.

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 dépassez le seuil choisi, mettez à l'échelle votre cluster en utilisant un type de nœud plus grand ou ajoutez des nœuds supplémentaires.

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 CurrConnections peut révéler un problème avec votre application ; vous aurez besoin d'analyser le comportement de l'application pour traiter ce problème.

Surveillance des métriques de cluster et de nœud CloudWatch

ElastiCache et CloudWatch sont intégrés afin que vous puissiez rassembler différentes métriques. Vous pouvez surveiller ces métriques à l'aide de CloudWatch.

Note

L'exemple d'interface de ligne de commande nécessite les outils de ligne de commande CloudWatch. Pour plus d'informations sur CloudWatch et pour télécharger les outils de développement, accédez à la [page produit CloudWatch](#).

Les procédures suivantes indiquent comment utiliser CloudWatch pour collecter les statistiques d'espace de stockage d'un cluster de cache au cours de l'heure qui vient de s'écouler.

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 en savoir plus sur les limites ElastiCache, veuillez consulter [AWS Service Limits](#) pour ElastiCache.

Surveillez les métriques CloudWatch cluster et nœud (Console)

Pour collecter les statistiques d'utilisation de l'UC d'un cluster de cache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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 gestion AWS, cliquez sur le nom d'un ou 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.

Une liste des métriques CloudWatch disponibles s'affiche au bas de la fenêtre de la console.

- d. Cliquez sur la métrique CPU Utilization.

Le console CloudWatch s'ouvre en affichant vos métriques 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 de cluster et de nœuds CloudWatch à l'aide de la CLI CloudWatch

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 de cluster et de nœuds CloudWatch à l'aide de l'API CloudWatch

Pour collecter les statistiques d'utilisation de l'UC d'un cluster de cache

- Appelez l'API CloudWatch `GetMetricStatistics` avec les paramètres suivants (notez que les dates de début et de fin sont affichées uniquement à titre d'exemple ; vous devez les remplacer par les dates 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`

Exemple

```
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>

Surveillance des événements ElastiCache avec Amazon SNS

Lorsque des événements importants se produisent sur un cluster de cache, 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 notifications ElastiCache Amazon SNS](#)
- [Affichage des événements ElastiCache](#)
- [Notifications d'événements Amazon SNS](#)

Gestion des notifications ElastiCache Amazon SNS

Vous pouvez configurer ElastiCache pour envoyer des notifications pour des événements de cluster importants 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](#).

Ajout d'une rubrique Amazon SNS

Les sections suivantes vous expliquent comment ajouter une rubrique Amazon SNS à l'aide de la console AWS, de la AWS CLI ou de l'API ElastiCache.

Ajout d'une rubrique Amazon SNS (console)

La procédure suivante vous indique comment ajouter une rubrique Amazon SNS pour un cluster.

 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 AWS Management Console et ouvrez la console ElastiCache 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 commande `modify-cache-cluster` de la AWS CLI.

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 plus d'informations, consultez [modify-cache-cluster](#).

Ajout d'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`

Exemple

```
https://elasticache.amazonaws.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 plus d'informations, consultez [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 d'Amazon SNS à l'aide de l'AWS Management Console

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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.
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
```

Activation et désactivation des notifications Amazon SNS (API ElastiCache)

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 :

Exemple

```
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 événements ElastiCache

ElastiCache consigne les événements relatifs 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 récupérer facilement les événements à partir du journal en utilisant la console ElastiCache, la commande AWS CLI de la `describe-events` ou l'action `DescribeEvents` de l'API ElastiCache.

Les procédures suivantes vous montrent comment afficher tous les événements ElastiCache qui ont eu lieu au cours des dernières 24 heures (1 440 minutes).

Affichage des événements ElastiCache (Console)

La procédure suivante affiche les événements à l'aide de la console ElastiCache.

Pour afficher les événements à l'aide de la console ElastiCache

1. Connectez-vous à la AWS Management Console et ouvrez la console ElastiCache 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.

Dans l'écran Événements, chaque ligne de la liste représente un événement, affichant sa source, son type (`cache-cluster`, `cache-parameter-group`, `cache-security-group` ou `cache-subnet-group`), l'heure GMT de l'événement, ainsi qu'une description de cet événement.

A l'aide du Filtre, vous pouvez choisir d'afficher tous les événements ou uniquement ceux d'un type spécifique dans la liste des événements.

Affichage des événements ElastiCache (AWS CLI)

Pour générer une liste des événements ElastiCache à l'aide de la 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 des événements ElastiCache (API ElastiCache)

Pour générer une liste des événements ElastiCache à l'aide de l'API ElastiCache, utilisez l'action `DescribeEvents`. Vous pouvez utiliser des paramètres facultatifs pour contrôler le type et la période des événements répertoriés, le nombre maximal d'événements à répertorier, etc.

Le code suivant répertorie les 40 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 :

- Une seule rubrique peut être configurée pour des notifications ElastiCache.

- Le compte AWS qui possède la rubrique Amazon SNS doit être le même compte 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. Toutefois, l'état de la rubrique de la console ElastiCache s'affiche comme inactif, ce qui dissocie effectivement la rubrique du cluster lorsque ElastiCache envoie des messages à la rubrique.

- La rubrique Amazon SNS doit se trouver dans la même région que le cluster ElastiCache.

Événements ElastiCache

Les événements ElastiCache suivants déclenchent des notifications Amazon SNS. Pour de plus amples informations sur les détails de l'événement, veuillez consulter [Affichage des événements ElastiCache](#).

Nom de l'événement	Message	Description
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	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 disponibles	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	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 : <i>cluster-name</i>	Un ou plusieurs paramètres du cluster de cache ont été modifiés.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	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 de réseau incompatible	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Il y a eu une tentative de lancement d'un nouveau cluster de cache dans un Virtual Private Cloud (VPC) inexistant.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionnement du cluster de cache réalisé avec succès.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	Le dimensionnement du cluster de cache a échoué.

Nom de l'événement	Message	Description
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	L'un des événements suivants s'est produit : <ul style="list-style-type: none">• La liste des groupes de sécurité de cache autorisés à accéder au cluster de cache a été modifiée.• Un ou plusieurs nouveaux groupes de sécurité EC2 ont été autorisés à accéder aux groupes de sécurité de cache associés au cluster de cache.• Un ou plusieurs nouveaux groupes de sécurité EC2 ont été déçus de leur autorisation d'accès par les groupes de sécurité de cache associés au cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache a détecté que l'hôte exécutant un nœud de cache est endommagé ou inaccessible, et a commencé à remplacer le nœud de cache.</p> <div data-bbox="1068 495 1507 806"><p> Note</p><p>L'entrée DNS pour le nœud de cache remplacé n'est pas changée.</p></div> <p>Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Toutefois, certaines bibliothèques client de cache peuvent cesser d'utiliser le nœud de cache même après qu'ElastiCache ait remplacé le nœud de cache ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.</p>

Nom de l'événement	Message	Description
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache a détecté que l'hôte exécutant un nœud de cache est endommagé ou inaccessible, et a terminé le remplacement du nœud de cache.</p> <div data-bbox="1068 541 1507 856"><p> Note</p><p>L'entrée DNS pour le nœud de cache remplacé n'est pas changée.</p></div> <p>Dans la plupart des cas, vous n'aurez pas besoin d'actualiser la liste des serveurs pour vos clients lorsque cet événement se produit. Toutefois, certaines bibliothèques client de cache peuvent cesser d'utiliser le nœud de cache même après qu'ElastiCache ait remplacé le nœud de cache ; dans ce cas, l'application doit actualiser la liste des serveurs lorsque cet événement se produit.</p>

Nom de l'événement	Message	Description
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	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 (uniquement Redis)	ElastiCache:CertificateRenewalComplete	Le certificat Amazon d'une autorité de certification a été renouvelé avec succès.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Le groupe de réplication a bien été créé.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	La suppression d'un cluster de cache et de tous les nœuds de cache associés est terminée.
ElastiCache:FailoverComplete (uniquement Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	Basculement vers un nœud du réplica réussi.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Le nombre de réplicas dans le cluster a été augmenté.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Le processus d'ajout de réplicas à votre cluster a commencé.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Le remplacement d'un nœud de votre cluster qui était prévu a été annulé.
ElastiCache:NodeRescheduled	ElastiCache:NodeRescheduled : <i>cluster-name</i>	Le remplacement d'un nœud de cluster a été reprogrammé dans le créneau indiqué dans la notification. Pour en savoir plus sur les mesures que vous pouvez prendre, consultez Remplacement des nœuds de cache pour Memcached .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	Un nœud du cluster doit être remplacé pendant le créneau décrit dans la notification. Pour en savoir plus sur les mesures que vous pouvez prendre, consultez Remplacement des nœuds de cache pour Memcached .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nœud de cache a été supprimé du cluster de cache.

Nom de l'événement	Message	Description
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	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 (uniquement Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Un instantané de cache s'est terminé avec succès.
ElastiCache:SnapshotFailed (uniquement Redis)	SnapshotFailed : <i>cluster-name</i>	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 événements ElastiCache](#)

Journalisation des appels d'API Amazon ElastiCache avec AWS CloudTrail

Amazon ElastiCache est intégré avec AWS CloudTrail, un service qui enregistre les actions effectuées par un utilisateur, un rôle ou un service AWS dans Amazon ElastiCache. CloudTrail capture tous les appels d'API pour Amazon ElastiCache en tant qu'événements, y compris les appels de la console Amazon ElastiCache et les appels de code à des opérations d'API Amazon ElastiCache. Si vous créez un journal d'activité, vous pouvez activer la livraison continue des événements CloudTrail dans un compartiment Amazon S3, y compris pour les événements relatifs à Amazon ElastiCache. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). Avec les informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à Amazon ElastiCache, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur et la date de la demande, ainsi que d'autres détails.

Pour en savoir plus sur CloudTrail, consultez le [AWS CloudTrailGuide de l'utilisateur](#) .

Informations Amazon ElastiCache dans CloudTrail

CloudTrail est activé dans votre compte AWS lors de la création de ce dernier. Lorsqu'une activité se produit dans Amazon ElastiCache, elle est enregistrée au sein d'un événement CloudTrail avec d'autres événements de services AWS dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez [Affichage des événements avec l'historique des événements CloudTrail](#).

Pour un enregistrement continu des événements de votre compte AWS, y compris les événements relatifs à Amazon ElastiCache, créez un journal d'activité. Un journal de suivi permet à CloudTrail de livrer des fichiers journaux dans 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 d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser et agir sur les données d'événements collectées dans les journaux CloudTrail. Pour en savoir plus, consultez les ressources suivantes :

- [Présentation de la création d'un journal d'activité](#)
- [Intégrations et services pris en charge par CloudTrail](#)
- [Configuration des Notifications de Amazon SNS pour CloudTrail](#)

- [Réception des fichiers journaux CloudTrail de plusieurs régions](#) et [Réception des fichiers journaux CloudTrail de plusieurs comptes](#)

Toutes les actions Amazon ElastiCache sont enregistrées par CloudTrail et sont documentées dans la [Référence d'API ElastiCache](#). À titre d'exemple, les appels vers les actions `CreateCacheCluster`, `DescribeCacheCluster` et `ModifyCacheCluster` génèrent des entrées dans les fichiers journaux CloudTrail.

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 les autorisations utilisateur root ou IAM .
- Si la demande a été effectuée avec des autorisations de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez l'[élément `userIdentity` CloudTrail](#).

Présentation des entrées des fichiers journaux Amazon ElastiCache

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

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action `CreateCacheCluster`.

```
{
  "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 présente une entrée de journal CloudTrail qui illustre DescribeCacheCluster action. Notez que pour tous les appels décrire les appels Amazon ElastiCache Describe (Describe*), la section ResponseElements est supprimée et apparaît en tant que 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 journal CloudTrail qui enregistre une action ModifyCacheCluster.

```

{
  "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"
}
```

Quotas pour ElastiCache

Votre compte AWS dispose de quotas par défaut, anciennement appelés limites, pour chaque service AWS. 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 afficher les quotas pour ElastiCache, ouvrez la boîte de dialogue [Service Quotas Console \(Console Service Quotas\)](#). Dans le volet de navigation, choisissez Services AWS et sélectionnez 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](#).

Les quotas de votre compte AWS concernant ElastiCache sont les suivants :

Ressource	Par défaut
Caches sans serveur par région	40
Nœuds par région	300
Nœuds par cluster	40
Groupes de paramètres par région	300
Groupes de sécurité VPC par région	50
Groupes de sous-réseaux par région	300
Sous-réseaux par groupe de sous-réseaux	20

Référence

Les rubriques de cette section présentent l'utilisation de l'API Amazon ElastiCache et de la section ElastiCache de la AWS CLI. Sont également présentés les messages d'erreur et les notifications de service courants.

- [Utilisation de l'API ElastiCache](#)
- [Référence d'API ElastiCache](#)
- [Section ElastiCache de la référence AWS CLI](#)
- [Message d'erreur Amazon ElastiCache](#)
- [Notifications](#)

Utilisation de l'API ElastiCache

Cette section fournit des descriptions de tâches et décrit comment utiliser et mettre en œuvre les opérations Elasticache. Pour une description complète de ces opérations, consultez le document [Référence d'API Amazon ElastiCache](#)

Rubriques

- [Utilisation de l'API Query](#)
- [Bibliothèques disponibles](#)
- [Applications de dépannage](#)

Utilisation de l'API Query

Paramètres Query (Requête)

Ces demandes basées sur Query HTTP sont des demandes HTTP qui utilisent le verbe HTTP GET ou POST et un paramètre Query appelé Action.

Chaque demande Query doit inclure certains paramètres communs pour gérer l'authentification et la sélection d'une action.

Certaines actions demandent des listes de paramètres. Ces listes sont spécifiées en utilisant la notation `param.n`. Les valeurs de `n` sont des nombres entiers à partir de 1.

Authentification de demande Query

Vous pouvez envoyer uniquement des demandes Query via HTTPS, et vous devez inclure une signature dans chaque demande Query. Cette section explique comment créer la signature. La méthode décrite dans la procédure suivante est appelée signature version 4.

Voici les étapes de base utilisées pour authentifier les demandes à AWS. Ce processus suppose que vous êtes enregistré avec AWS et disposez d'un ID de clé d'accès et d'une clé d'accès secrète.

Processus d'authentification des requêtes

1. L'expéditeur crée une demande à AWS.
2. L'expéditeur calcule la signature de la demande, un hachage avec clé pour un code HMAC (code d'authentification d'une empreinte cryptographique de message avec clé) utilisant une fonction de hachage SHA-1, comme défini dans la prochaine section de cette rubrique.
3. L'expéditeur de la demande envoie les données de la demande, la signature et l'ID de clé d'accès (l'identifiant de la clé d'accès secrète utilisée) à AWS.
4. AWS utilise l'identifiant de la clé d'accès pour rechercher la clé d'accès secrète.
5. En appliquant le même algorithme utilisé pour calculer la signature dans la demande, AWS génère une signature à partir des données de la demande et de la clé d'accès secrète.
6. Si la signature correspond, la demande est considérée comme authentique. Si la comparaison échoue, la demande est rejetée, et AWS renvoie une réponse d'erreur.

Note

Si une demande contient un paramètre `Timestamp`, la signature calculée pour la demande expire 15 minutes après sa valeur.

Si une demande contient un paramètre `Expires`, la signature expire au moment spécifié par le paramètre `Expires`.

Pour calculer la signature de la demande

1. Créez la chaîne de requête de base que vous utiliserez à une étape ultérieure de la procédure :

- a. Triez les composants de la chaîne de requête UTF-8 par nom de paramètre disposé selon l'ordre naturel des octets. Les paramètres peuvent provenir de l'URI GET ou du corps POST (lorsque le type de contenu est application/x-www-form-urlencoded).
 - b. URL-encodez le nom et les valeurs du paramètre en appliquant les règles suivantes :
 - i. Ne pas URL-encoder les caractères que le RFC définit comme autorisés. Les caractères autorisés sont A à Z, a à z, 0 à 9, le trait d'union (-), le trait de soulignement (_), le point final (.) et le tilde (~).
 - ii. %-encodez tous les autres caractères avec %XY, où X et Y représentent les caractères hexadécimaux 0 à 9 et les lettres majuscules A à F.
 - iii. %-encodez les caractères UTF-8 étendus dans la forme %XY%ZA....
 - iv. %-encodez le caractère espace en %20 (et non pas en +, comme le font les schémas d'encodage courants).
 - c. Utilisez le symbole équivalent (=) (ASCII caractère 61) pour séparer les noms de paramètres codés de leurs valeurs codées, même si la valeur du paramètre est vide.
 - d. Séparez les paires nom-valeur en insérant une esperluette (&) (code ASCII 38).
2. Créez la chaîne de connexion en appliquant la grammaire suivante (le « \n » représente une nouvelle ligne ASCII).

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

Le composant HTTPRequestURI est le composant du chemin absolu HTTP de l'URI menant jusqu'à la chaîne de requête de demandes sans cependant l'inclure. Si le composant HTTPRequestURI est vide, utilisez une barre oblique (/).

3. Définissez un HMAC conforme à RFC 2104 à l'aide de la chaîne que vous venez de créer, votre clé d'accès secrète comme clé et l'algorithme haché SHA256 ou SHA1.

Pour plus d'informations, consultez <https://www.ietf.org/rfc/rfc2104.txt>.

4. Convertissez la valeur qui est générée en Base64.
5. Incluez la valeur comme la valeur du paramètre Signature dans la demande.

Par exemple, voici un exemple de demande (sauts de ligne ajoutés pour plus de clarté).

```
https://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 les kits de développement logiciel (SDK) pour les développeurs qui veulent créer des applications à l'aide d'API langage spécifique au lieu de l'API Query. Ces kits de développement logiciel (SDK) offrent des fonctions de base (non présentes dans les API), telles que l'authentification de demande, les nouvelles tentatives de demande et la gestion des erreurs ; celles-ci vous permettent de démarrer plus facilement. Des kits de développement logiciel et des ressources supplémentaires sont disponibles pour les langages de programmation suivants :

- [Java](#)
- [Windows et .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Pour plus d'informations sur les autres langages, consultez [Exemples de code et bibliothèques](#).

Applications de dépannage

ElastiCache fournit des erreurs spécifiques et descriptives pour vous aider à résoudre vos problèmes tout en interagissant avec l'API ElastiCache.

Récupération d'erreurs

Généralement, vous souhaitez que votre application vérifie si une demande a généré une erreur avant de passer du temps à traiter les résultats. Le moyen le plus simple de déterminer si une erreur s'est produite est de rechercher un nœud `Error` dans la réponse de l'API ElastiCache.

La syntaxe XPath fournit une méthode simple pour rechercher la présence d'un nœud `Error` et récupérer le code et le message d'erreur. L'extrait de code suivant utilise Perl et le module

XML::XPath pour déterminer si une erreur s'est produite lors d'une demande. Si une erreur s'est produite, le code imprime le premier code et message d'erreur dans la réponse.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Conseils pour le dépannage

Nous vous conseillons d'utiliser les processus suivants pour diagnostiquer et résoudre les problèmes avec l'API ElastiCache.

- Vérifiez qu'ElastiCache s'exécute correctement.

Pour ce faire, il vous suffit d'ouvrir une fenêtre de navigateur et d'envoyer une demande de requête au service ElastiCache (par exemple, <https://elasticache.amazonaws.com>). Une erreur `MissingAuthenticationTokenException` ou de serveur interne 500 confirme que le service est disponible et qu'il traite les demandes.

- Vérifiez la structure de votre demande.

Chaque opération ElastiCache possède une page de référence dans la Référence de l'API ElastiCache. 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.

Un forum de discussion ElastiCache permet de chercher des solutions à vos problèmes auprès d'autres utilisateurs qui ont les déjà rencontrés. Pour consulter le forum, rendez-vous à l'adresse

<https://forums.aws.amazon.com/>.

Configuration de l'interface de ligne de commande ElastiCache

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 envisagez d'utiliser la AWS CLI pour ElastiCache.

Important

L'interface de ligne de commande (CLI) d'Amazon ElastiCache ne prend en charge aucune amélioration d'ElastiCache après la version d'API 2014-09-30. Pour utiliser les nouvelles fonctionnalités d'ElastiCache à partir de la ligne de commande, utilisez [l'interface de ligne de commande AWS](#).

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 suppose que vous pouvez travailler dans un environnement Windows ou Linux/UNIX. Les outils de ligne de commande Amazon ElastiCache fonctionnent également sur Mac OS X, qui est un environnement basé sur UNIX ; toutefois, aucune instruction propre à Mac OS X n'est incluse dans ce manuel.

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 utilisons également **\$** pour désigner une commande propre à Linux/UNIX et **C:\>** pour une commande propre 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 les clients JRE pour différentes plateformes, y compris Linux/UNIX et Windows, consultez [Java SE Downloads](#).

Définition de la variable Home de Java

Les outils de ligne de commande dépendent 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 ils comportent des scripts 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 dépendent 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 les numéros de version/édition), et il contient les 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 d'accès AWS et de la clé d'accès secrète fournies avec votre compte AWS. 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 :

```
AWSAccessKeyId=<Write your AWS access ID>  
AWSSecretKey=<Write your AWS secret key>
```

⚠ Important

Sous UNIX, limitez les autorisations au propriétaire du fichier d'identification :

```
$ chmod 600 <the file created above>
```

Une fois le fichier d'identification configuré, vous devez définir la variable d'environnement `AWS_CREDENTIAL_FILE` de sorte que les outils ElastiCache puissent trouver vos informations.

Pour régler 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 API inclus avec l'interface de ligne de commande ElastiCache prennent en charge les variables suivantes :

- `EC2_REGION` – région AWS à utiliser.

- `AWS_ELASTICACHE_URL` — URL à utiliser pour l'appel du service. Il n'est pas nécessaire d'indiquer un point de terminaison régional différent si `EC2_REGION` est spécifié ou si le paramètre `--region` est transmis.

Les exemples ci-après illustrent comment définir la variable d'environnement `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
```

Message d'erreur Amazon ElastiCache

Les messages d'erreur suivants sont renvoyés par Amazon ElastiCache. Vous pouvez recevoir d'autres messages d'erreur qui sont renvoyés par ElastiCache, par les autres services AWS, ou par Memcached. Pour obtenir une description des messages d'erreur à partir de sources autres que ElastiCache, consultez la documentation de la source qui a généré le message d'erreur.

- [Cluster node quota exceeded](#)
- [Customer's node 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, effectuez votre demande à l'aide du [Formulaire de demande de nœuds Amazon ElastiCache](#).

Pour plus d'informations, consultez [Limites Amazon ElastiCache](#) 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, effectuez votre demande à l'aide du [Formulaire de demande de nœuds Amazon ElastiCache](#).

Pour plus d'informations, consultez [Limites Amazon ElastiCache](#) dans Référence générale d'Amazon Web Services.

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 [Mise à l'échelle ElastiCache pour Memcached](#).

Notifications

Cette rubrique couvre les notifications ElastiCache 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 des fonctions ou fonctionnalités améliorées ou nouvellement créées d'ElastiCache.

Notifications générales ElastiCache

Actuellement, il n'y a pas de notifications ElastiCache en attente qui ne sont pas spécifiques au moteur.

Notifications ElastiCache for Memcached

Les notifications ElastiCache suivantes sont spécifiques au moteur Memcached.

Notifications ElastiCache for Memcached spécifiques

- [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:



- 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 l'utilisateur ElastiCache pour Memcached ElastiCache . Pour recevoir les notifications sur les mises à jour de cette documentation, vous pouvez vous abonner au Flux RSS.

Mises ElastiCache à jour récentes de Memcached

Modification	Description	Date
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 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	27 novembre 2023

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 .

[Support ElastiCache pour Memcached 1.6.17](#)

ElastiCache pour Memcached 18 janvier 2023
supporte désormais Memcached 1.6.17. Il inclut des corrections de bogues cumulatives à partir de la version [Memcached 1.6.12](#). Pour plus d'informations, consultez [Memcached version 1.6.17](#).

[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 les connexions IPv4, uniquement les connexions IPv6 ou les connexions IPv4 et IPv6 (double pile). IPv6 est pris en charge pour les charges de travail utilisant le moteur Memcached à partir de la version 1.6.6 sur toutes les instances créées sur le [système Nitro](#). Il n'y a pas de frais supplémentaires pour l'accès ElastiCache via IPv6. Pour plus d'informations, 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 ultérieures. Pour plus d'informations, consultez la section [Chiffrement ElastiCache en transit \(TLS\)](#).

26 mai 2022

[ElastiCache prend désormais en charge PrivateLink](#)

AWS PrivateLink vous permet d'accéder aux opérations d'ElastiCache API de manière privée sans passerelle 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 \(AWS PrivateLink\) pour Redis ou ElastiCache Amazon API et points de terminaison VPC d'interface \(\)](#) pour Memcached. AWS PrivateLink

24 janvier 2022

[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 Memcached 1.6.6.](#)

ElastiCache pour Memcached 12 novembre 2020
supporte désormais Memcached 1.6.6. Il inclut les correctifs de bogue cumulatifs des versions Memcached 1.5.16. Pour plus d'informations, veuillez consulter [Memcached version 1.6.6](#).

[ElastiCache est désormais disponible sur AWS Outposts](#)

[AWS Outposts apporte des](#) 8 octobre 2020
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 de plus amples informations, veuillez consulter [Utilisation d'Outposts](#) pour Redis ou [Utilisation d'Outposts](#) pour Memcached.

[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 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 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 utilise 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

[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 Standard T3 d'Amazon EC2 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. Pour plus d'informations, consultez [Types de nœuds pris en charge](#).

12 novembre 2019

[ElastiCache pour Memcached 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. Ainsi, les interruptions de service seront réduites au minimum, en particulier pendant les pics de trafic, et vous serez assuré de vous trouver en conformité avec les mises à jour de sécurité. Pour plus d'informations, consultez la section [Mises à jour en libre-service sur Amazon ElastiCache](#).

9 octobre 2019

[Support ElastiCache pour Memcached 1.5.16](#)

ElastiCache pour Memcached supporte désormais Memcached 1.5.16. Il inclut les correctifs de bogue cumulatifs des versions [Memcached 1.5.14](#) et [Memcached 1.5.15](#). Pour plus d'informations, veuillez consulter [Memcached version 1.5.16](#).

6 septembre 2019

ElastiCache Offres d'instances réservées standard : initiale partielle, initiale complète et aucune mise 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
Support ElastiCache pour Memcached 1.5.10	ElastiCache pour Memcached supporte désormais Memcached 1.5.10. Il prend également en charge les paramètres no_modern et inline_ascii_resp. Pour en savoir plus, consultez Memcached version 1.5.10.	14 novembre 2018
Nouvelle structure du guide de l'utilisateur	Le guide de ElastiCache l'utilisateur unique est maintenant restructuré afin qu'il existe des guides d'utilisation distincts pour Redis (pour le guide de l'utilisateur de Redis) et ElastiCache pour Memcached (ElastiCache pour le guide de l'utilisateur de Memcached). La structure de documentation de la section AWS CLI Command Reference : elasticache et celle de l'Amazon ElastiCache API Reference restent inchangées.	20 avril 2018

Le tableau suivant décrit les modifications importantes apportées au guide de l'utilisateur ElastiCache pour Memcached ElastiCache .

Modification	Description	Date de modification
Prise en charge de la région Asie-Pacifique (Osaka-Local).	<p>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 invitation uniquement. Pour plus d'informations, consultez les ressources suivantes :</p> <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	12 février 2018
Prise en charge de l'UE (Paris).	<p>ElastiCache support supplémentaire pour la région UE (Paris). Pour plus d'informations, consultez les ressources suivantes :</p> <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	18 décembre 2017
Prise en charge de la région Chine (Ningxia)	<p>Amazon ElastiCache a ajouté le support pour la région de Chine (Ningxia). Pour plus d'informations, consultez les ressources suivantes :</p> <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	11 décembre 2017
Prise en charge des rôles liés aux services	<p>Cette version ElastiCache ajoute la prise en charge des rôles liés au service (SLR). Pour plus d'informations, consultez les ressources suivantes :</p>	7 décembre 2017

Modification	Description	Date de modification
	<ul style="list-style-type: none"> • Utilisation des rôles liés à un service pour Amazon ElastiCache • Configurez vos autorisations (nouveaux ElastiCache utilisateurs uniquement) 	
Prise en charge des types de nœud R4	<p>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 :</p> <ul style="list-style-type: none"> • Types de nœuds de cache pris en charge • Paramètres propres au type de nœud Memcached 	le 20 novembre 2017
Rubrique Modèles de connexion	<p>ElastiCache la documentation ajoute une rubrique couvrant les différents modèles d'accès à un ElastiCache cluster dans un Amazon VPC.</p> <p>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 .</p>	24 avril 2017
Prise en charge de Memcached 1.4.34	<p>ElastiCache ajoute le support de Memcached version 1.4.34, qui intègre un certain nombre de correctifs aux versions antérieures de Memcached.</p> <p>Pour plus d'informations, consultez les notes de mise à jour de Memcached 1.4.34 sur Memcached on. GitHub</p>	10 avril 2017

Modification	Description	Date de modification
Prise en charge de Memcached 1.4.33	<p>ElastiCache ajoute le support pour la version 1.4.33 de Memcached. Pour plus d'informations, consultez les ressources suivantes :</p> <ul style="list-style-type: none">• Memcached Version 1.4.33• Paramètres ajoutés Memcached 1.4.33	20 décembre 2016
Prise en charge pour la région UE Ouest (Londres)	<p>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 :</p> <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	13 décembre 2016
Prise en charge pour la région Canada (Montréal)	<p>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 :</p> <ul style="list-style-type: none">• 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	<p>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 :</p> <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	1 novembre 2016

Modification	Description	Date de modification
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 : <ul style="list-style-type: none">• Régions prises en charge• Types de nœuds de cache pris en charge	17 octobre 2016
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 : <ul style="list-style-type: none">• Types de nœuds de cache pris en charge• Paramètres propres au type de nœud Memcached	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 : <ul style="list-style-type: none">• Types de nœuds de cache pris en charge• Paramètres propres au type de nœud Memcached	27 juin 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

Modification	Description	Date de modification
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 informations, veuillez consulter Tutoriels et vidéos ElastiCache .	12 février 2016
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
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'algorithme de hachage, de nouvelles commandes et divers correctifs. Pour de plus amples informations, veuillez consulter Notes de mise à jour de Memcached .	27 août 2015
Prise en charge de découverte 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 Compilation du code source pour le client de cluster ElastiCache pour PHP .	29 juillet 2015
Nouveau sujet : Accès ElastiCache depuis l'extérieur AWS	Ajout d'un nouveau sujet sur l'accès aux ElastiCache ressources depuis l'extérieur AWS. Pour plus d'informations, consultez la section Accès ElastiCache depuis l'extérieur AWS .	9 juillet 2015

Modification	Description	Date de modification
Messages de remplacement de nœud ajoutés	<p>ElastiCache ajoute trois messages relatifs au remplacement planifié des nœuds <code>NodeReplacementScheduled</code>, <code>NodeReplacementRescheduled</code>, et <code>NodeReplacementCanceled</code>.</p> <p>Pour plus d'informations et pour connaître les mesures que vous pouvez prendre lorsqu'il est prévu de remplacer un nœud, consultez ElastiCache la section Notifications d'événements Amazon SNS.</p>	11 juin 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
Support pour la AWS GovCloud région (ouest des États-Unis)	ElastiCache ajoute le support pour la région AWS GovCloud (US-Ouest) (<code>us-gov-west-1</code>).	29 janvier 2015
Prise en charge de la région Europe (Frankfurt)	ElastiCache ajoute le support à la région Europe (Francfort) (<code>eu-central-1</code>).	19 janvier 2015
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 d'API Amazon ElastiCache avec AWS CloudTrail .	15 septembre 2014

Modification	Description	Date de modification
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 de moteur à l'aide de groupes de paramètres .	11 septembre 2014
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 de moteur à l'aide de groupes de paramètres .	1 juillet 2014
Découverte automatique PHP	Ajouter la prise en charge de la découverte automatique PHP version 5.5.	13 mai 2014
Prise en charge d'Amazon Virtual Private Cloud (VPC) par défaut	Dans cette version, ElastiCache est entièrement 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 Sécurité des VPC Amazon et ElastiCache .	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

Modification	Description	Date de modification
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 Sécurité des VPC Amazon et ElastiCache .	20 décembre 2012
Découverte automatique des nœuds de cache et nouvelle version de moteur de cache	<p>ElastiCache permet la découverte automatique des nœuds de cache, c'est-à-dire la capacité des programmes clients de déterminer automatiquement tous les nœuds de cache d'un cluster, ainsi que d'établir et de maintenir des connexions avec tous ces nœuds.</p> <p>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'évolutivité et des performances, ainsi que plusieurs correctifs. Plusieurs nouveaux paramètres de cache peuvent être configurés. Pour de plus amples informations, veuillez consulter Configuration des paramètres de moteur à l'aide de groupes de paramètres.</p>	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.	5 avril 2012

Modification	Description	Date de modification
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.